Database Design Description (DBDD)
Phase 2

United States National Data Center (US NDC)

CONTRACT NO. F33657-99-C-4722
SAIC-01/3047 (REV A) CDRL B024

17 July 2002

Prepared for:

DET 3 ASC

1030 South Highway A1A

Patrick AFB, FL 32925-3002

Prepared by:

Science Applications International Corporation
Global Systems Division

1227 S. Patrick Drive, Suite 110
Satellite Beach, FL 32937

Revision History

DATE	REVISION	DESCRIPTION	AUTHOR	AUTHORIZED BY
24 Aug 01	-	Pre-release	T. Lyons	T. Ingalls
8 Feb 02	-	Phase 2 Initial Release	T. Lyons	T. Ingalls
23 Apr 02	-	Phase 2 Final Release	T. Lyons	T. Ingalls
17 May 02	-	Phase 2 Final Update to Incorporate Government Comments and Include Alt US NDC	T. Lyons	T. Ingalls
17 Jul 02	A	Phase 2 Final Update to Incorporate Government Comments on Alt US NDC	T. Lyons	T. Ingalls

Science Applications International Corporation
Global Systems Division
1227 S. Patrick Drive, Suite 110
Satellite Beach, FL 32937

Database Design Description (DBDD) Phase 2, Revision A: United States National Data Center (US NDC)

Primary Author: T. Lyons

Contributing Authors: D. Irons, B. MacRitchie, N. Putvinskaya

Table of Contents

List of Figures

List of Tables

1. Scope

1.1 Identification

This Database Design Description (DBDD) defines the database design for the United States National Data Center (US NDC) Phase 2 System and the US NDC Phase 2 Training System. The US NDC Phase 2 System is located at the Air Force Technical Applications Center (AFTAC) at Patrick Air Force Base (PAFB), Florida, and the US NDC Phase 2 Training System is located at Goodfellow Air Force Base (GAFB), Texas. The database requirements for the US NDC Phase 2 System and the US NDC Phase 2 Training System are defined in the US NDC Phase 2 *System Requirements Document (SRD)*. See SAIC 02/3009, System/Subsystem Design Description Phase 2 (*US NDC)* for the traceability matrix.

Note: For the remainder of the document, the term US NDC databases will be understood to mean US NDC Phase 2 System databases.

1.2 Database Overview

The US NDC consists of several Oracle Relational Database Management System (RDBMS) instances that support operations, development, and sustainment activities. An instance is often referred to as a database. The difference between a database and an instance is described in Section 3.4.1. Each database runs on a different hardware platform, known as a host. The use of separate hosts reduces the risk of one database affecting the performance of another.

The data acquisition and data processing databases contain the tables necessary to support routine daily operations, including all import and export of raw data, all routine automatic processing, and all interactive analysis of data. Records remain in these databases for three to six weeks, which keeps them compact while providing an adequate buffer in case problems are identified and need to be corrected prior to migrating the data to the archive databases.

The archive databases contains all tables that may be of long-term interest and, therefore, worth preserving. Data migrate regularly from the data acquisition and data processing databases to the archive databases.

The archive databases contain all tables that may be of long-term interest and, therefore, are worth preserving. Data migrates regularly from the data acquisition and data processing databases to the archive databases..

The Alternate US NDC (Alt US NDC) duplicates the functionality of the US NDC, including the data acquisition, data processing, and archiving databases. The Alt US NDC databases incorporate the same schema as their US NDC counterparts. During normal mission operations at AFTAC, raw data is forwarded from the US NDC to the Alt US NDC and acquired on the Alt US NDC databases. Data processing results produced on the US NDC are copied to the Alt US NDC data processing database via Oracle Replication. The contents of the classified and

unclassified archive databases are also copied to their Alt US NDC counterparts via Oracle Replication. When the Alt US NDC assumes the mission, data is acquired on Alt US NDC databases from primary sources and data processing results are produced locally at the Alt US NDC. At the same time, the archiving databases at the Alt US NDC begin accumulating historical data from the data acquisition or data processing databases at the Alt US NDC via the archiving and data migration processes.

The purpose of the US NDC Phase 2 Training System is to train United States Air Force (USAF) personnel to analyze data collected and processed by the US NDC Phase 2 System. The US NDC Phase 2 Training System has one database used to support instructor-based training. The database on the US NDC Phase 2 Training System was designed to be almost identical to the US NDC Phase 2 data processing database. Because these databases are so similar, it is implicit that further references to the US NDC Phase 2 data processing database also apply to the US NDC Phase 2 Training System database.

The origin of the US NDC Phase 2 database schema is rooted in several legacy and operational systems. The Center for Seismic Study (CSS) Version 3.0 database design (Anderson, et al. 1990) defined the core tables in the CSS databases. It supported the extension of the core schema with application-specific tables. Since the definition of the core schema, a few extensions have been added to accommodate new functionality. The CSS added extensions in 1991 and 1993 to accommodate new functionality for the Intelligent Monitoring System (IMS) [see *IMS Extensions to the Center Version 3 Database* (Swanger, et al. 1993)]. AFTAC extensions were added to the core schema in 1996 to support AFTAC requirements. These extensions were incorporated into the AFTAC Distributed Subsurface Network (ADSN), which was the predecessor to the US NDC. Most of these extension tables have since been determined to be redundant; that is, the information contained within them can be determined from combinations of data contained in other tables and have been removed. In addition, in the past certain tables were added to the US NDC Phase 1 System and ADSN databases in anticipation of future usage which never materialized. These unused tables have been removed from the US NDC database design.

The US NDC databases described in this document contain new tables, which were added to the schema to support new features and improvements to existing functions. One major improvement is the generalized management of amplitude measurements and improved knowledge representation for regionalized magnitude calibration. The motivation and principles of the new representation is described in *Recommendations for the US NDC Data Management System(DMS)*, *SAIC-96/1099*, (Sereno, 1996). Several more tables were added to the databases to support new features, which include:

- New data acquisition format Continuous Data 1.1 (CD-1.1)
- New representation of amplitudes and magnitudes
- Hydroacoustic azimuth estimation
- New signal processing quality control (QC) functions

A major enhancement to the US NDC System accomplished as part of Phase 2 is an infrastructure upgrade. The operating system (OS) was upgraded to Sun Solaris 8 from Sun Solaris 2.6 and the RDBMS was upgraded from Oracle Server Enterprise Edition 7.3.4 to Oracle Server Enterprise Edition 8.1.7.

The upgrade from 7.3.4 to 8.1.7 provided the opportunity to take advantage of several new features to the RDBMS available in version 8.1.7. The new features that were considered for incorporation into the design of the US NDC databases are as follows:

- Increased security through single sign on and user/schema separation
- Consistency of query performance through the use of the Optimizer Plan Stability
- Improved RDBMS performance and reduced tablespace fragmentation through the use of locally managed tablespaces
- Improved garbage collection at the end of a transaction or session through the use of Session/Transaction level Temporary Tables
- Reduction in time to access archived wfdisc data through the use of partitioning
- Increased control over server resources through the Database Resource Management tool
- More flexible backup and recovery procedures through the use of Recovery Manager (RMAN)
- Easier database administration and maintenance through the use of Oracle Enterprise Manager (OEM)
- Improved performance of applications and ad-hoc queries through the use of alternative types of indexing such as bitmap and reverse key

The US NDC Phase 2 System is being sponsored by the AFTAC Nuclear Treaty Monitoring Directorate (AFTAC/TT), which identified the system requirements for the project. The project is being acquired under the guidance and control of Detachment (Det) 3, Aeronautical Systems Center (ASC), Air Force Material Command. The system delivered under this project is used by personnel of the AFTAC Directorate of Operations. The system is supported by the AFTAC Directorate of Logistics and Systems (LS).

The primary operating site for the US NDC Phase 2 System is the AFTAC facility at PAFB, Florida. The US NDC Phase 2 System will replace the existing US NDC Phase 1 system at AFTAC. A second US NDC Phase 2 System is planned for installation at GAFB, Texas shortly after installation of the system at AFTAC. It is to be used as an alternate site, ready to take over operations on short notice should, for any reason, the primary system become unavailable to support operations.

All relevant documents are listed in Section 2 of this document.

1.3 Document Overview

This document is prepared in accordance with (IAW) Data Item Description (DID) DI-IPSC- 81437A for a DBDD as tailored in Contract Data Requirements List (CDRL) B024.

- Section 1 provides the scope of this document, which includes a paragraph for identification, database overview, and this paragraph.
- Section 2 provides a list of referenced documents.
- Section 3 summarizes the database-wide design decisions for the databases.
- Section 4 provides detailed database design including descriptions of the conceptual, logical, internal, and physical models developed during the design process.
- Section 5 refers to the computer software component (CSC) documentation for detailed descriptions of the software units that access the databases.
- Section 6 refers to the traceability matrix located in *SAIC 02/3009*, *System/Subsystem Design Description (SSDD) Phase 2 Build 1 (US NDC).*
- Section 7 provides notes, to include an acronym list.
- Appendix A contains the table descriptions.
- Appendix B contains the column descriptions.
- Appendix C contains the view descriptions.
- Appendix D contains a list of accounts and tables.

1.4 Conventions

Table 1 provides the typographical conventions used in this document.

Table 1. Typographical Conventions

ELEMENT	APPEARANCE	EXAMPLE
Database table Database table and columns, when written in the dot notation	Bold	**dataready** **prodtrack.status**
Database columns Processes, software units, and libraries Titles of documents	Italics	*status* *ARS*, *libpar* *GA Subsystem Software*
Value of a key or component of a key	Courier font	`orid`
Database accounts and database names	All capital letters	GLOBAL LOOKUP
SQL*Plus statements or commands	Underline	Select

2. Reference Documents

SAIC-96/1063, United States National Data Center (US NDC) Database Schema Reference, Version 2.3, 17 January 2001

SAIC-00/3057, International Data Centre (IDC) Database Schema, IDC5.1.1, Rev 3, November 2001

SAIC-02/3009, System/Subsystem Design Description (SSDD) Phase 2 Build 1 (US NDC), 10 April 2002

SAIC-93/1001, Generic Database Interface (GDI) User Manual, 5 April 1994

SAIC-96/1099, Recommendations for the US NDC Data Management System(DMS), May, 1996

SAIC Internal Memorandum, Amplitude/Magnitude Schema Upgrade, January 1998

System Requirements Document (SRD) for the United States National Data Center (Phase 2), Document Number 1001901

System Requirements Document (SRD) for the United States National Data Center (Phase 2), Revision 0, Document Number 1002348, 7 September 2001

AFTAC Extensions to the CSS 3.0 Database Schema, AFTAC, January 1996 (in revision)

Operations Annex 2: GSE Database, GSETT-3 Working Group on Operations (WGO), March 1996

AFTAC Extensions Database Design Document, Version 1.1, October 1992

IMS Extensions to the Center Version 3 Database, February 1991

Center for Seismic Studies Version 3 Database Schema Reference Manual, September 1990

Oracle8i Application Developer's Guide -- Fundamentals, Release 2 (8.1.6), A76939-01, December 1999

Oracle Enterprise Manager (OEM), A75684-01, November 1999

Oracle Recovery Manager (RMAN), A76990-01, December 1999

Oracle8i Utilities Manual (8.1.6), A76955-01, December 1999

Oracle8i Installation Guidance Release 3 for Sun SPARC Solaris (8.1.7), A85417-01, December 1999

Oracle Call Interface (OCI) Programmer's Guide (8.1.6), A76975-01, December 1999

Oracle8i Concepts, Release 2 (8.1.6), A76965-01

*SQL*Plus User's Guide and Reference (8.1.6), A75664-01, October 1999*

PL/SQL User's Guide and Reference (8.1.6), A77069-01, December 1999

Proposal for Alternate US NDC, 17 May 2000

3. Database-wide Design Decisions

The following subsections describe database-wide design decisions and how those design decisions apply to the US NDC Phase 2 System database design.

Note: For the remainder of the document, the US NDC Phase 2 System will be referred to as the US NDC System.

3.1 System Architecture

The US NDC System architecture is a simplification, based on advances in the relevant computer technology, of the Phase 1 System architecture. It replaces the existing collection of 12 servers with four multi-domain servers. Multi-domain servers allow collections of processors within the server to be segregated into domains, which can function like multiple independent servers within a single chassis. In the US NDC, this permits allocation of separate domains to operate the various database instances required. More detailed descriptions of the US NDC System architecture can be found in *SAIC-02/3009*.

The US NDC multi-domain servers are each provided with mirrored disk arrays for the exclusive use of the domains which operate the Oracle databases. The disk arrays are configured with Veritas Volume Manager to define volumes that are striped across groups of disks in the array. When configured in this manner, the Volume Manager distributes the database file load evenly across the disks in the group. This technique relieves the database manager of the need to assign specific datafiles to specific disk spindles in an attempt to balance the load manually.

3.2 Database System Environment

The US NDC databases are designed to operate in a client/server environment. This architecture was chosen for many reasons, but mainly because the underlying architecture of the US NDC System is also client/server. In this environment, the database servers manage data shared by client processes. For the most part, client processes execute either on separate compute servers (data acquisition and automatic processing) or on desktop workstations (interactive processing). The clients provide application interfaces to the shared data on the servers.

A major component of the US NDC System is the collection of database instances managed using the Oracle Server Enterprise Edition version 8.1.7 software. Oracle 8.1.7 is the most recent version of the Oracle 8i Server that is compatible with the IDC R3 software baseline, the foundation for the US NDC System. Oracle 8.1.7 is also the latest version of the Oracle Server product that is approved by the Defense Information Infrastructure Common Operating Environment (DII COE) program as of the publication date of this document.

3.3 Use of New Oracle 8i Features

Oracle 8i includes a variety of optional new features that were considered in the database design process. Some were included in the Phase 2 design. Other new features that had been anticipated to be useful were found unsuitable for the use anticipated and were rejected for the US NDC database design.

3.3.1 Locally Managed Tablespaces

In previous generations of Oracle Server, management of disk space in the datafiles associated with tablespaces was handled in the data dictionary. With Oracle 8i, the option exists to allow each tablespace (except the SYSTEM tablespace) to manage its own space allocation via a bitmap in each datafile associated with the tablespace. Whenever an extent is allocated or freed for reuse, Oracle updates the bitmap to reflect the new status of the disk blocks assigned to the extent in question. Locally managed tablespaces have the following advantages over dictionary managed tablespaces:

- Local management minimizes rollback activity associated with dictionary updates
- Local management automatically tracks adjacent free space, eliminating the need to coalesce free extents

Oracle Corporation has already given notice that a future release of Oracle Server will remove the dictionary-managed tablespace capability for user defined tablespaces from the product. At that time, local space management will become the standard for user-defined tablespaces. Oracle is encouraging users to convert their databases to use locally managed tablespaces before that future release. Accordingly, the US NDC databases are being converted to use locally managed tablespaces.

3.3.2 Temporary Tables

An Oracle temporary table is a very specific concept introduced in Oracle 8i and is well described in the manuals *Oracle 8i Concepts* and *Oracle8i Application Developer's Guide*. Basically, temporary tables are database tables created during a session to hold session private data for either the duration of the session or the duration of a single transaction within the session. A session is equivalent to a user log on. Each application logs on to the database and thus has its own session. Temporary tables can NOT be seen by other sessions. If the table is not deleted during the session, it is automatically deleted by the server when the session ends (that is, the user or application logs off) or when the transaction ends (that is, a commit or rollback is issued), as appropriate.

For a single session or transaction, temporary tables may be used just as permanent tables are used. Triggers and indexes may be created on them. Views may be created on the temporary tables or on joins between them and the permanent tables. One difference between temporary and permanent tables is that temporary tables use temporary segments. Space is allocated on the first insert statement rather than pre-allocated at the time the table is created. Another difference

is that there are no locks on the table. Locks are not needed since only a single session can see or modify the data.

The use of the term temporary table in the US NDC Phase 1 System design pre-dates Oracle temporary tables. It refers to tables that are created to hold non-permanent data that is to be communicated to another application. The communication model used predominantly between US NDC applications is an interprocess communication (IPC) message containing references to data in the database. The IPC message does not contain the actual data, only the references to it (that is, *orid*s, *arid*s, and other keys or identifiers). The naming of US NDC database temporary tables follows a specific format:

TMP$_<application>_<tablename>_<uniqueid>

The format allows the tables to be identified and cleaned up automatically by the application or clean up scripts.

By definition, Oracle temporary tables cannot be used for the purpose above since they cannot be seen by more than one session. A few applications, the *Global Association (GA)* for example, use the database to store intermediate processing results (the **assoc_ga**, **origerr_ga**, and **origin_ga** tables). However, Oracle temporary tables cannot be used for this either because the data are used by multiple applications/sessions [example given (e.g.), *GAassoc, GAconflict*].

Generally, US NDC applications keep intermediate results and data in memory unless they need to be communicated to other applications. It may be that future software development efforts will find Oracle temporary tables applicable. However, none of the existing applications currently use the database in this manner.

3.3.3 Database Resource Management

Database Resource Management is a newly added database administration feature with Oracle 8i. It allows the database administrator to control the allocation of processing resources to jobs and sessions executing within the Oracle database instance. The administrator can assign levels of priority and percentages of central processing unit (CPU) resources to resource consumer groups, which can then be assigned to sessions. On the US NDC System, this mechanism is used to favor automated processing and interactive sessions by analysts and evaluators over other types of access such as ad hoc querying via the read-only accounts provided for such use. This will limit the risk that long running ad hoc queries could reduce the responsiveness of the US NDC databases to operational requirements.

3.3.4 Oracle Enterprise Manager (OEM)

The OEM is a comprehensive Graphical User Interface (GUI)-based product for database administration of a network of databases of any size and complexity desired. It is based on a three tier architecture including the following:

- A GUI-based client console with management applications for common database administration tasks

- A middleware Oracle Management Server which maintains a repository of system information and processes system management tasks from the console(s)
- Oracle Intelligent Agents on the servers with databases to be administered

The OEM can provide a variety of graphically depicted information about the status of databases that is not readily obtainable via structured query language (SQL) commands. It can provide service monitoring for pre-set events and automatically respond to situations needing attention. It provides a Job Scheduling System to automate routine database administration tasks. The Intelligent Agents can be configured to operate autonomously and perform scheduled tasks even if the console or management server is unavailable.

The OEM needs a repository to catalog the information that it collects. Since the repository requires a database instance, a separate minimal database instance called RCAT is created specifically to hold this information.

3.3.5 Recovery Manager (RMAN)

The RMAN is a new backup and recovery tool introduced as the standard for managing backup and recovery operations with Oracle 8i. It provides a complete GUI, which is fully integrated with OEM. RMAN has several advantages over the backup method in use by the Phase 1 US NDC databases, including the following:

- RMAN automatically detects fractured blocks thus eliminating the need to place tablespaces in hot backup mode.
- RMAN catalogs and manages all the backup artifacts (e.g., backup files and archive logs) that are produced and uses the catalog to facilitate recovery
- RMAN can perform incremental backups of databases
- RMAN can eliminate the need to use OS-level commands as part of a backup or recovery operation

Accordingly, RMAN has been adopted as the standard backup/recovery tool for US NDC databases. RMAN offers the option of using a dedicated database as the repository for its catalog or using the control file in the target database as the repository. Since use of a dedicated database is recommended, the US NDC System uses the RCAT database instance, which is also used for the OEM repository, as the catalog for all backup/recovery information for the US NDC databases.

3.3.6 Optimizer Plan Stability

This Oracle 8i feature permits the database administrator to manage the performance of common queries executed on a database. Ordinarily, when a query is executed on a database, the oracle cost-based optimizer generates a query plan for the execution of that query at the time that the query is submitted. The query plan is, essentially, a best guess by the optimizer at a plan for using indexes to accelerate the execution of the query, as opposed to performing full table scans every time. Since the query plan is developed anew each time the query is submitted, the query

plan, and thus the query execution time, can change significantly, based on the state of the indexes at query time. Optimizer Plan Stability attempts to minimize this variability in query execution by storing query plans for commonly executed queries in a system data structure in the database called a query plan outline. The Oracle cost-based optimizer can be configured to search the query plan outlines before attempting to formulate a new cost-based query plan in response to a request to execute a query. This ensures that commonly executed queries use the same query plan every time. While this is not always appropriate, there have been instances where commonly used queries in the US NDC System have had their execution plans changed radically by the optimizer for no apparent reason. In such cases, the traditional technique for obtaining relief from such occurrences has been to modify the application to apply hints to the query in question. Since many US NDC applications construct their queries on the fly, insertion of hints into the queries is not always practical. Limited use of Optimizer Plan Stability for such commonly constructed queries will help alleviate the problem of queries not being consistent in their execution times.

3.3.7 Advanced Security Option

The Advanced Security Option was considered as a possible tool for separating the US NDC database users from the permanent schema elements that hold US NDC data processing results. This is desirable because when an interactive user logs in to an account that owns critical schema elements such as database tables, the user has the inherent capability to alter the structure of the schema, perhaps inadvertently. This can have catastrophic results for the operation of the US NDC System, bringing some or all automatic processing activities to a halt and possibly resulting in the loss of data.

The Advanced Security Option has a feature called User/Schema Separation that allows users to have access to application specific schemas. While this appears to be a suitable technique for solving this problem, it has a limitation that makes it unsuitable. Users must first be created in a global naming directory outside of any database. These are called enterprise users and they are given enterprise roles also defined in the directory. The databases available to enterprise users are also registered in the naming directory and application schemas are associated with enterprise roles. When an enterprise user logs into a database, his enterprise role is checked to validate his access to an application schema. Unfortunately, the enterprise user at this point does not have a schema of his own and thus has given up the ability to create tables. This is unacceptable to the US NDC applications since many of them create and drop tables in the course of passing data between processes. For this reason, use of the Advanced Security Option was rejected for the US NDC System. An alternative technique, which does achieve a measure of user/schema separation, is described in Section 3.5.2.

3.4 Database Servers and Databases

The US NDC System receives raw waveforms from both classified and unclassified sensors distributed worldwide. The waveforms arrive on the Classified and Unclassified Systems, respectively. Waveforms from unclassified sources are forwarded to the Classified System.

Once gathered, the waveforms are stored on the filesystem and the associated waveform description (wfdisc) records are cataloged in a database for immediate and delayed processing. The system manages the raw waveforms, using geophysical algorithms for the detection and identification of man-made seismic events. All analysis results are gathered and stored in a database. There is a requirement to provide storage for continuous processing of recently received data and long term storage of historical data and processing results. This requirement led to the decision to have separate database instances configured to support these two disparate requirements. One database, supporting continuous processing, is configured according to the guidelines for On-line Transaction Processing (OLTP). The other database, supporting historical data, is configured as a data warehouse. The OLTP database is referred to as the OPSDB and the historical or archive database is referred to as the ARCHDB. An OPSDB and an ARCHDB reside on both the Unclassified and Classified Systems.

3.4.1 Overview of Databases and Instances

The terms database and instance are often used interchangeably, but there is an important distinction between the two. A database is a set of data. Data in a database is stored in tables. The US NDC databases follow the Relational Data Model. Relational tables are defined by their columns, also referred to as attributes. Data is stored as rows in the tables. Tablespaces are used to provide a logical mapping within the RDBMS to physical storage managed by the OS. A tablespace is a logical, internal data storage structure. The physical data structure is called a datafile, which is visible in the UNIX filesystem. Objects in the database (e.g., tables) are assigned to tablespaces, which are mapped to datafiles.

An instance is composed of memory structures and background processes that access the database files. The primary memory structure is the System Global Area, which maintains a complex set of shared data structures by which the background processes communicate with each other. These background processes perform various tasks essential to proper operation of the instance. For example, the SMON process monitors the instance to ensure that temporary objects used during a completed transaction have been cleaned up. It also coalesces contiguous free space in the tablespaces to reduce internal fragmentation. PMON, DBWR, LGWR, CKPT, ARCn, RECO, SNPn, LCKn, Dnnn and Snnn are examples of some of the other background processes. Additional detail on the internal structure of a database instance and the processes that support it may be found in the manual *Oracle 8i Concepts, Release 2 (8.1.6)*.

For the remainder of this document, the US NDC instances and databases are referenced together as databases.

The RDBMS needs several types of datafiles to operate and maintain the database. These files are:

- Tablespace datafiles
- Redo (transaction) logs
- Control files
- Trace files

- Alert logs

In the event that a database hardware or software failure results in a loss of data, the database administrator must be able to recover the lost data. Two additional types of datafiles are maintained for this purpose:

- Archived transaction logs
- Backup database files

Under certain recovery scenarios, a control file can be used to facilitate recovery.

3.4.2 Database Server Configuration

The physical configuration of each database server supports the function of the database it hosts. Disk configurations vary from machine to machine, based on the amount of storage space needed. For example, the archive database servers have more disk space than the data acquisition and processing database servers because longer data storage is required on the archive machines.

Memory parameters, set in the OS kernel before the database is created, determine how shared memory is used by the database. The *Oracle 8i Installation Guide* provides guidelines for setting these parameters based on the type of database and the number of anticipated database transactions.

As mentioned in Section 3.1, disks on the database servers are managed by the Veritas Volume Manager. While it would be possible to create a single, giant volume encompassing all the disk storage, this could permit a single out-of-control process to consume all disk storage, bringing processing to a halt. Instead, several logical volumes are created to hold the different types of datafiles and other file structures needed to operate the databases.

Each database has at least one datafile per tablespace. Each database has more than one copy of the control file. It is written to each disk as a trace file by a UNIX CRON job that runs on the server. The alert logs are contained in the same directory on each machine.

The database filesystem configuration on all US NDC database servers follows the basic tenets of the Optimal Flexible Architecture (OFA) recommended in the *Oracle8i Installation Guide*. The software is installed in an Oracle UNIX account where the home directory is designated as the UNIX environment variable ORACLE_BASE. The Oracle software is installed at $ORACLE_BASE/product/rel, which is designated ORACLE_HOME. The file structure supports multiple databases, each designated by its system identifier designated ORACLE_SID. UNIX symbolic links pointing to the datafiles associated with a particular database are gathered under $ORACLE_BASE/oradata/$ORACLE_SID. This use of symbolic links allows the distribution of datafiles to various hardware disk partitions to be hidden from the Oracle instance itself, making database administration and maintenance tasks easier to perform.

3.5 Database Accounts

The primary logical structure by which a user or application accesses the database is the database account. Normally, an account consists of an associated schema and a username and password, which must be transmitted from the client to the database server when objects in the schema are to be accessed. The schema associated with an account is composed of different objects, including the following:

- Tables
- Indexes
- Synonyms
- Links
- Views
- Triggers
- Sequences
- Packages
- Procedures
- Jobs

The US NDC databases have a system of database accounts that support the data acquisition and pipeline processing modes of operation. Automatic processes and interactive users perform their functions by connecting to the database via one of these accounts. The raw data coming in from the field is cataloged in one account. Pipeline processes perform a sequence of analysis functions, the results of which are collected in one of several pipeline accounts. A succession of pipeline accounts collects the results of each stage of processing and makes the data available to the next stage in the pipeline. Each pipeline consists of one or more stages of automated processing which produce preliminary results as well as one or more stages of interactive processing in which trained analysts review and adjust the results of the automatic processing as their judgment dictates. Several of these processing pipelines are incorporated into the US NDC System design. In addition, there are special accounts in which processes perform data archiving or performance monitoring. There is also a reference account, which contains configuration data describing the international network of stations which supply data to the US NDC System. The accounts are described in Section 4 in terms of the role they play in data acquisition, data processing, data archiving, performance monitoring and database maintenance functions of the US NDC System.

3.5.1 Synonyms

Oracle synonyms are aliases for Database Objects (dbObjs) such as tables, views, and PL/SQL functions and stored procedures. Synonyms can be used for masking the name or location of an object and to simplify SQL statements. PL/SQL functions and procedures can be stored and maintained in a single globally available account with synonyms providing access to the

functions and procedures to all other accounts. Synonyms can also be used to provide security via user/schema separation as described in Section 3.5.2. Synonyms can be owned by individual accounts or created as public synonym, in which case they are visible to all accounts in the database.

The US NDC database design uses synonyms to simplify SQL statements, to achieve user/schema separation, and to configure the pipelines. The use of synonyms places the burden of configuring the accounts into pipelines on the database rather than on the programs that use the accounts. A program logs onto a single database account and sees all the objects it needs to perform its function. In actuality, however, many of the objects are synonyms pointing to objects in other accounts. The application need not understand the complexities of the pipeline to perform its function. Furthermore, a user logging onto an account to perform ad hoc queries, perhaps through SQL*Plus, will also be able to easily see all the data pertinent to the functional purpose of that account.

Figure 1 illustrates the use of synonyms by the US NDC pipeline accounts. The specific design of the US NDC database is that the accounts for a pipeline (that is, DETPRO, SOCCPRO, and AL1 in the Global pipeline) have synonyms to tables in the LOOKUP and GLOBAL accounts. For instance, all the pipeline accounts have a lastid synonym pointing to the **lastid** table in the GLOBAL account. As far as the applications are concerned, they are accessing a table called **lastid** in the current account, but in actuality they all access a single **lastid** table in the GLOBAL account. Similarly the pipeline accounts have synonyms to the **site**, **affiliation** and other reference tables in the LOOKUP account.

The pipeline stage accounts also use synonyms to access the accounts of previous stages. For example, SOCCPRO contains synonyms to tables in DETPRO and AL1 contains synonyms to tables in both SOCCPRO and DETPRO. By convention, these synonyms have a name of the form in_<tablename>, (that is, in_arrival, in_assoc), to distinguish them from other tables and synonyms pointing to the LOOKUP and GLOBAL accounts. The programs producing results to be stored in the SOCCPRO account log onto SOCCPRO. They access their input data from the DETPRO account through the in_ synonyms. Programs never write data to the in_ tables.

The AL1 account carries the use of synonyms one step further, with the out_ synonyms. The out_ synonyms point to tables in the current account (that is, AL1). This emphasizes the fact that the purpose of the account is for an analyst to interactively review the automatic results from previous stages of the pipeline and produce a corresponding but separate set of reviewed tables and data. For instance, AL1 has in_arrival, in_assoc, in_origin, etc., as well as out_arrival, out_assoc, out_origin, etc. The out_ tables take precedence over the in_ tables; however, the automatic and reviewed results are both available.

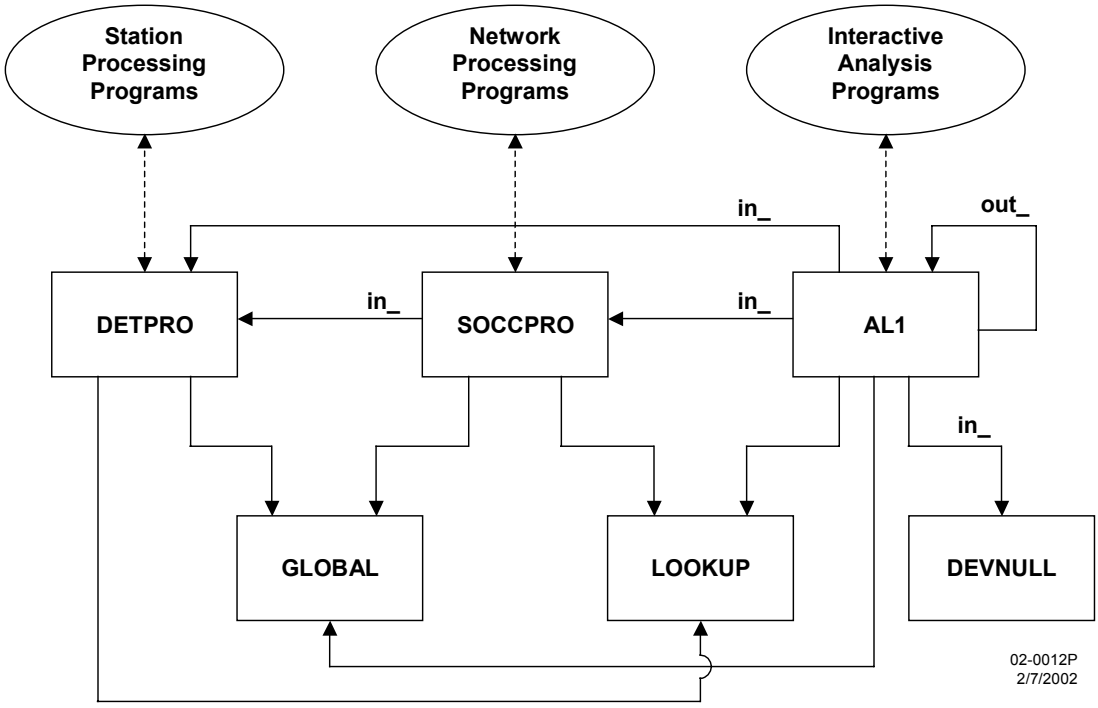


Figure 1. Synonyms in US NDC Database Accounts

3.5.2 User/Schema Separation

In the ordinary course of operation, the schemas of the US NDC data acquisition and pipeline processing accounts are fixed and immutable, while the data contained within the schema elements is constantly being updated. If an interactive user logs in to an account that owns schema elements, that interactive user has the capability to alter or drop schema elements, perhaps inadvertently. Such an inadvertent action could prove catastrophic to US NDC System operations. Accordingly, an approach has been devised to give the interactive user full access to the data in the data acquisition and processing accounts while removing the ability to alter the structure of the schema associated with the processing stage.

Each data acquisition and pipeline processing stage will actually have three accounts associated with it. One account will own the permanent schema elements associated with that stage of processing. A second account is granted insert, update and delete permissions on the schema elements (in effect, a read/write account). A third account is granted only select permissions on the schema elements (in effect, a read-only account). The following example illustrates the naming convention adopted for these accounts:

- AL1_OWNER -- owner of schema elements for AL1 processing stage
- AL1 -- interactive user performing AL1 processing stage activities

- AL1_RO -- interactive user needing read-only access to AL1 processing stage results via ARS, SQL*Plus, or other interactive access methods

In this approach, the AL1 and AL1_RO accounts will have synonyms pointing to the actual AL1 schema elements owned by the AL1_OWNER account. This structure of accounts and synonyms is replicated for all the data acquisition and pipeline processing accounts.

3.6 Database Inputs and Outputs

The purpose of this section is to describe design decisions regarding the inputs the US NDC databases will accept, and the outputs they are expected to produce.

Input and output are accomplished through one of several interfaces. The interfaces to the US NDC databases can be categorized as command-line interfaces, programmatic interfaces or GUIs. The Oracle applications SQL*Plus and SQL*Loader provide command-line access to the contents of a database. Examples of programmatic interfaces are the Oracle Call Interface (OCI), Java Database Connectivity (JDBC), PL/SQL and custom Application Programming Interfaces (APIs). The OEM and RMAN provide access to the databases through GUIs.

The following paragraphs describe the interfaces to the US NDC databases in terms of how they are used on the US NDC System.

3.6.1 Command Line Interfaces

Oracle Server Enterprise Edition version 8.1.7 includes many tools and utilities that provide access to a database through a command-line interface. These interfaces are used to perform a variety of tasks such as data manipulation and data definition, loading the contents of a text file into a database or exporting the contents of a database or account for import into another database or account. The US NDC System uses the following command-line Oracle utilities: SQL*Plus, SQL*Loader, Import, and Export.

3.6.1.1 SQL*Plus

SQL*Plus provides users with a command-line interface to the US NDC databases through ad hoc SQL statements entered into a UNIX shell. SQL statements are also embedded in UNIX shell scripts that are used to automate database administration and maintenance procedures. These scripts are run at scheduled intervals using the UNIX CRON utility.

When users connect to a database through a SQL*Plus session, there are two initialization files read by the database server processes that set the user's environment. These files are called glogin.sql and login.sql. The location of the glogin.sql file is determined by the value of $ORACLE_HOME environmental variable and the location of the login.sql file is determined by the value of the $SQLPATH environmental variable. The location in the filesystem to which these variables are set is determined by how the user's environment is set before he or she initiates an SQL*Plus session. The default parameters in glogin.sql that were set during the software installation are not modified. The login.sql file has been modified to set column

formats, variable definitions, and to execute SQL scripts needed to configure the accounts database environment (see the *SQL*Plus User's Guide and Reference 8.1.6* for more information).

3.6.1.2 SQL*Loader

The MigrateData application uses SQL*Loader to load data from American Standard Code for Information Interchange (ASCII) text files into database tables. SQL*Loader requires two ASCII text input files to load data, a file with a .dat extension that contains the data to be loaded and another file that describes the structure of the table in which the data is loaded. This file has a .ctl extension. A custom US NDC application, *sqluldr*, provides a complimentary function to SQL*Loader; it extracts data from database tables and writes them to ASCII text files. In order to extract data from a database table, *sqluldr* requires an ASCII text file that specifies the column format definitions for the output file; the aftac_column_formats.txt file is an example of one of these files. For more information on *sqluldr* see the man page and for more information on SQL*Loader see the *Oracle8i Utilities Manual*.

3.6.1.3 Import and Export Utilities

The *TrainingExport* and *TrainingImport* applications use the Import and Export utilities to export data from the databases at AFTAC to the US NDC Phase 2 Training System database. These utilities are also used for occasional database administration and maintenance tasks and for development purposes. They can be run either interactively or non-interactively from the command line. In interactive mode, the Oracle Server prompts the user for input. See the *Oracle8i Utilities Manual* for more information.

3.6.2 Programmatic Interfaces

US NDC custom applications, third-party applications and Commercial Off-the-Shelf (COTS) products are programmatic interfaces to the US NDC databases. Each of these applications and products have an API which they use to access data in the databases through an intermediate interface called the OCI.

3.6.2.1 Oracle Call Interface (OCI)

The OCI is Oracle's custom API that allows the user to create applications that use the native procedures or function calls of a third-generation programming language to access an Oracle database server and control all phases of SQL statement execution. OCI supports the datatypes, calling conventions, syntax, and semantics of a number of third-generation languages including C, C++, COBOL, and FORTRAN. For more information on the OCI, see the *Oracle Call Interface Programmer's Guide 8.1.6*.

3.6.2.2 Generic Database Interface (GDI)

US NDC applications use a GDI called *libgdi* to interact with the OCI and the Oracle databases. The GDI provides database connection management, SQL query execution, and data handling. It insulates applications from differences between versions of OCI and from differences between Oracle's and other vendor's database API. The design has separate front-end and back-end components, thus allowing it to support multiple vendors or version at one time. The US NDC Phase 1 System used a back-end that interfaced with the Oracle 7 version of OCI. The Oracle 8 version of OCI is entirely different; however, applications were ported to *libgdi* by the Prototype International Data Centre (PIDC) and migrated seamlessly to Oracle 8 through a new *libgdi* back-end, also developed by the PIDC.

The front-end comes in several variations. A dbObj provides a self-describing C structure to manage any data corresponding to an arbitrary SQL query or set of database columns. An array of structures interface provides arrays of simple C structures specific to database tables or predefined queries. The *libsdi* library provides an interface for S-PLUS applications; it allows a user to interactively execute an ad hoc database query at the S-PLUS prompt, then transparently transfers query results into S-PLUS, where they can be manipulated with S-PLUS functions.

3.6.2.3 PL/SQL

PL/SQL is an Oracle extension to the SQL that combines the data manipulation power of SQL with the data processing power of procedural languages. PL/SQL can be installed in an Oracle server or in an application development tool such as Oracle Forms or Oracle Reports. On the US NDC System, PL/SQL is installed in the Oracle Server. The PL/SQL blocks can be embedded within an OCI program or can be compiled and stored separately in the databases as subprograms. Once compiled and stored in the data dictionary, a subprogram is a schema object, which can be referenced by any number of applications connected to the database. Stored subprograms can also be defined within a package with other stored subprograms.

The US NDC databases use PL/SQL packages to perform daily database administration and maintenance tasks such as gathering object statistics which are used during query processing to improve performance. PL/SQL triggers are also used to update certain tables, which support waveform archiving.

The *PL/SQL User's Guide and Reference 8.1.6* provides more information on the use and features of PL/SQL.

3.6.2.4 Perl

Perl applications access the US NDC databases through the Perl Database Independent Interface (DBI). The DBI defines a set of functions, variables, and conventions that provide a consistent database interface independent of the actual database being used. The DBI provides a standard interface and framework in which the Perl drivers operate. Drivers are Perl modules that implement support for a specific type of database. Drivers contain implementation of the

DBI methods written using the private interface function of the corresponding database. The driver used to access data in the US NDC databases is DBD::Oracle.

The US NDC System uses Perl applications to perform a variety of data manipulation and data definition tasks. For example, *load_lookup* is a Perl application that updates the tables in the Lookup account with information from ASCII text files that are stored in the US NDC configuration tree. Several of the US NDC data acquisition utilities are also written in Perl (see the Perl man pages for more information on the DBI and DBD::Oracle).

3.6.2.5 Java

Java applications access the database through JDBC. JDBC is an open standard developed by Sun Microsystems for connecting Java applications to relational databases.

In addition to supporting the standard JDBC API, Oracle drivers have extensions to support Oracle-specific datatypes and to enhance performance. See the *Oracle8i JDBC Developer's Guide and Reference* for more information on JDBC.

The only US NDC Java application that accesses the database is the new Configuration Browser, which displays field site configuration data.

3.6.3 Graphical User Interface (GUI)

The US NDC System uses two Oracle GUIs, OEM and RMAN, to access and manipulate data in the databases. These GUIs are provided with the Oracle Server Enterprise Edition software.

3.6.3.1 Oracle Enterprise Manager (OEM)

The OEM is a system management tool that provides an integrated solution for centrally managing a heterogeneous environment. Combining a graphical console, Oracle Management Servers, Oracle Intelligent Agent, common services and administrative tools, OEM provides a comprehensive systems management platform for managing Oracle products. From the OEM console, the database administrator can perform the following tasks:

- Centrally administer, diagnose, and tune multiple databases
- Manage Oracle products and services other than databases
- Effectively monitor and respond to the health of Oracle family of products and third-party services
- Schedule activities on multiple-nodes at varying time intervals
- Monitor networked services
- Streamline database administration and maintenance by combining and organizing databases and other services into logical administrative groups.

OEM is used on the US NDC System to augment existing manual and automated database administration and maintenance tasks. See the *Oracle Enterprise Manager Release 2.1* documentation for more information on OEM.

3.6.3.2 Recovery Manager (RMAN)

RMAN is an Oracle tool that allows for the back up, copy, restoration, and recovery of datafiles, control files, and archived transaction logs. RMAN can be invoked as a command line utility or the RMAN GUI, which is integrated with the OEM console GUI, can be used.

RMAN automates many of the backup and recovery functions that were formerly performed by UNIX shell scripts run as CRON jobs. For more information on RMAN, see the *Oracle8i Recovery Manager User's Guide and Reference*.

3.7. Unique Identifiers

Many tables in the US NDC database schema include a column referred to as an identifier that contains unique numbers as a primary or alternate key. Many applications may have read and write privileges for these tables and these applications may run in more than one US NDC schema account, so a method for ensuring the uniqueness of the identifier is required. For some tables, an Oracle sequence is used. For other tables, unique identifiers are obtained from the **lastid** table. When a sequence is used, Oracle guarantees the uniqueness of the number obtained. When the **lastid** table is used, however, the following procedures are necessary:

1. Obtain numbers from the **lastid** table by using select for update. This procedure places a lock on the **lastid** table, thereby blocking other applications that are also seeking to obtain a unique identifier for the same *keyname*. Applications built around *libgdi* use the *gdi_get_counter()* function to get new identifiers.
2. Immediately increment the *keyvalue* with an update query and commit the transaction. This procedure frees the lock allowing the next application to obtain a unique identifier.

There must be one definitive common **lastid** table visible to all applications operating in all accounts on the system. This common **lastid** table resides in the GLOBAL account. Numerous applications connecting to different accounts must have access to the **lastid** table. Therefore, update privileges on the **lastid** table are granted to a number of accounts.

In addition to the identifiers used for formation of primary keys, there is an additional identifier known as temp$object which is used to guarantee the uniqueness of table names for the temporary tables created by interactive applications such as *ARS* or *Discrim*. Access to the **lastid** table for the purpose of updating this identifier must be granted to the users of these applications in the read-only accounts to allow these applications to run.

3.8 Database Links

Database links allow database instances to be treated like a single, integrated database. A link tells Oracle how to get from one database to another and includes a communications protocol [such as Transmission Control Protocol/Internet Protocol (TCP/IP)], the name of the remote database host, the name of the remote database, a valid account name in the remote database, and the account password. Database links may be private, meaning that only a single user account may use the link, or public, meaning that any user account may use the link. Only private database links are used in the US NDC System. The only place database links are used in the US NDC is to enable Lookback processing to extract data from the archive for reprocessing.

3.9 Cloning

Several US NDC applications use clones of database schema tables as storage areas for intermediate data values during execution. The data in some of these tables is treated as temporary data and it is cleaned up by the application. Data stored in some of the other tables is treated as permanent data and it is copied into the archive database in the same manner as data contained in non-clone schema tables. The *GA* application uses the **assoc_ga**, **origerr_ga**, and **origin_ga** tables to store temporary values which are passed between the multiple instances of *GA* that run successively. The final *GA* instance that runs in any processing pass clears out this temporary data.

3.10 Database and Account Creation

All databases and their objects are created by means of UNIX shell scripts. There is a top-level shell script which calls a sequence of intermediate level shell scripts that perform the following tasks:

1. Create the database with only a SYSTEM tablespace
2. Add tablespaces and rollback segments as needed
3. Construct a data dictionary for the database
4. Configure the database to support replication
5. Install Java language support components in the database
6. Install online help information in support of SQL*Plus
7. Enable archive log mode of database operation
8. Establish default and temporary tablespaces
9. Create US NDC database accounts
10. Create US NDC database schemas

The first eight scripts in the above list are created initially using the Oracle Database Creation Assistant as called by the Oracle Universal Installer. These scripts are then edited by database

administration personnel to include US NDC System-specific configuration data necessary to the creation of the database. The first eight scripts create an empty database with OFA-compliant tablespaces and data files ready to accept US NDC database accounts and schema. The last two scripts in the above list create the US NDC accounts and schemas described in this document. The last of the intermediate scripts listed above calls a sequence of low level scripts for each account, each of which creates an instance of one of the objects in the schema associated with that account.

3.11 Table Creation and Tablespaces

Table creation and deletion is, in general, restricted to installation and maintenance tasks. The majority of application programs connecting to the databases do not create or drop tables. Notable exceptions are *ARS*, *MigrateData*, and *Discrim*, which create and drop tables as part of routine operations. Accounts where these applications are run have default tablespaces which are separate from the tablespaces that hold the permanent schema elements of the US NDC System. Tables not included in Appendix A of this document and not created by application software should not be created in the US NDC database accounts.

3.12 Indexes

Query performance is closely related to table indexing. Because the creation or dropping of arbitrary indexes can cause Oracle to choose different execution plans (resulting in possibly degraded performance), all indexes should be maintained with the same level of vigilance as the tables themselves. The following rules provide a general policy for indexes. Specific information for each index, including sizing information, is provided later in this document.

- All indexes associated with an account reside in a common tablespace, distinct from the tablespace used for the tables themselves
- All indexes are owned by the table owner. Other users (accounts) are prohibited from creating indexes on foreign tables. This can be accomplished by granting the create index privilege rather than the create any index privilege
- Each table has one unique index associated with the primary key
- Each table has one unique index associated with any alternate key
- Additional indexes chosen to facilitate query performance initially duplicate the indexes in the US NDC Phase 1 System. Further additional indexes are chosen as performance problems observed during integration testing dictate.

3.13 Data Migration, Archiving, and Purging

The US NDC System maintains an ARCHDB that corresponds to each of the data acquisition and data processing databases (OPSDB). The ARCHDB is a historical record of the processing results accumulated in the data acquisition and data processing database accounts.

The US NDC System uses *MigrateData*, an application that runs periodically as a CRON job to move data from one database to the other. *MigrateData* opens a connection to each database and performs its function according to a set of rules stored in tables in OPSDB. From the rules, *MigrateData* constructs SQL to select records in OPSDB tables, copies the records selected to corresponding ARCHDB tables, and verifies that the copy was correct and complete. At a later time, *MigrateData* executes another set of rules which select records in OPSDB tables for deletion after verifying that the range of records selected has been accurately migrated to the corresponding ARCHDB tables.

3.14 Data Integrity

In previous generations of the US NDC System, primary responsibility for data integrity belonged to the applications software. Applications ensured that column values were within appropriate limits and that duplicate keys and orphan records were not introduced. In the US NDC System, additional checks are performed by the RDBMS itself through the use of database constraints. This ensures the integrity of data entered by means other than applications software (e.g., ad hoc insert/update statements issued via SQL*Plus). The following paragraphs describe the data integrity constraints used in the US NDC database.

3.14.1 Data Type Checking

Before a value is inserted into the database, the RDBMS checks it to ensure that it is consistent with the Oracle data type for that field. Accordingly, selecting appropriate datatypes for all columns is very important.

3.14.2 Primary Keys

Primary keys are defined for virtually all tables in the US NDC database schema. Primary keys are columns, or combinations of columns, which uniquely identify records in a table. A primary key constraint results in a unique index on the primary key columns and prevents duplicate data from being introduced into the table. The primary key definition for each permanent table in the US NDC schema can be found in Appendix A.

The US NDC databases use Oracle primary keys. Prior generations of the US NDC System and predecessor systems did not use Oracle primary key constraints. However, most tables had unique indexes defined for the set of columns that made up the primary key. The indexes performed the identical function as primary keys would have performed.

3.14.3 Foreign Keys

In Appendix A, foreign keys are defined for most of the tables IAW the data models presented in Section 4 of this document. However, only a limited subset of the keys can be implemented as Oracle foreign key constraints in the database. In some cases, the entity relationship is too complex to be defined within the allowable syntax for foreign key constraint specifications. In other cases, the key relationship cannot be enforced as a constraint because the related records

are not produced in the order that would be demanded by the key and cannot be encapsulated in a single transaction. Nonetheless, the applications software has sufficient logic to ensure that the appropriate entity relationship is ultimately resolved according to the model.

3.14.4 Unique Indexes

In some cases, additional unique indexes are defined to ensure that uniqueness of alternate keys specified in the table definition of Appendix A is enforced by the database and to improve query performance.

3.14.5 Check Constraints

Check constraints are used on virtually all table columns in the US NDC database schema where the allowable range is less than the full range of values supported by the Oracle datatype. The US NDC applications follow a general policy of writing default or Not Applicable (NA) values when valid data is not available. The check constraint values are the union of the valid range and the NA value. Ranges and NA values are specified in the column definitions in Appendix B of this document. In a few cases, the range definition is too complex to be implemented as a check constraint. Such complex range definitions are included as information so that range checks can be implemented in the applications.

3.15 Database Maintenance

3.15.1 Routine Procedures

3.15.1.1 Automated Procedures

Datafiles associated with tablespaces are configured with the autoextend feature, which allows the datafile to grow up to specified limits.

3.15.1.1.1 CRON Jobs

Initially, the US NDC databases are configured with the same set of UNIX CRON jobs that are currently used by other US NDC Phase 1 databases. Some of these procedures may be replaced by facilities of OEM in the future. The jobs are executed in the Oracle UNIX account and produce electronic mail (e-mail) directed to that user which indicates whether the CRON job functioned normally or not. The mail system can be configured to forward that e-mail to any user in the database administration community.

3.15.1.1.2 Oracle Jobs

Oracle jobs are periodic procedures implemented in PL/SQL using the DBMS_JOBS package. In the US NDC databases, jobs are used for a variety of purposes:

- Schedule procedures which rebuild certain critical indexes
- Update statistics on the use of indexes which facilitates proper use of the cost-based optimizer
- Update tables associated with the archiving processes.

3.15.1.2 Manual Procedures

The alert log is monitored on a daily basis to determine if any error conditions have been recorded. The operator should monitor the status of jobs to verify they are operating properly. The operator monitors e-mail produced by UNIX CRON jobs that perform automated administration tasks to verify they are functioning normally.

3.15.2 Backups

In the US NDC Phase 1 System, backups were based on UNIX shell scripts executed as CRON jobs as described in Section 3.15.1.1.1. Recovery was a manual process, which involved using UNIX commands to retrieve and uncompress the backup files and place them in the appropriate locations on the filesystem and then using the Oracle Server Manager command line to recover the state of the database. In the US NDC System, both backup and recovery are managed by the RMAN utility. Comprehensive and incremental backups are taken periodically by RMAN, using the scheduling features of OEM. The ratio of incremental to comprehensive backups is determined experimentally once the database is fully integrated with the software. Oracle recommends that the size of incremental and comprehensive backups be compared. When the size of the incremental backup approaches half the size of the comprehensive backup, a new comprehensive backup should be scheduled.

3.16 Alternate United State National Data Center (Alt US NDC)

The Alt US NDC provides for the development of a functionally comparable US NDC located at GAFB, Texas. This system is able to assume the mission of the US NDC as circumstances dictate, in accordance with the requirements.

The Alt US NDC has a configuration of hardware, software, and databases that is identical to the configuration installed at the US NDC. In particular, the Alt US NDC databases have the exact same stet of accounts and schemas as their US NDC counterparts, as described in Section 4 of this document. The corresponding databases in the US NDC and Alt US NDC have the same database name and are distinguished from one another by the database domain portion of the global service name. The database domains are .USNDC and .ALTNDC. Thus, the OPSDB instance on each domain can be referenced as OPSDB.USNDC or OPSDB.ALTNDC, as needed.

During normal operations, the US NDC mission is performed at PAFB, Florida, and sensor data continues to be directly received only by the ADSN and US NDC equipment operating at PAFB. Since the Alt US NDC must be able to assume the US NDC mission at GAFB with little or no advanced notice, its data structures is constantly updated with data that is initially received or

produced at PAFB. Waveform data is continually forwarded from the US NDC to the Alt US NDC in CD-1.1 format via a Government-furnished Wide Area Network (WAN) between PAFB and GAFB. Thus, the Alt US NDC will already have this data when it begins to directly receive new sensor data upon US NDC mission transfer to GAFB.

Although forwarding CD-1.1 data accommodates a major portion of populating the Alt US NDC with mission data, there is other data (e.g., beams and database alphanumerics) that also must be synchronized between the US NDC and the Alt US NDC. Synchronization of alphanumeric data tables is accomplished through the use of Oracle Replication. Propagation of beams and their associated **wfdisc** and **wftag** entries is accomplished by means of newly developed custom software. More detail regarding the design decisions that produced the overall Alt US NDC architecture may be found in the *System/Subsystem Design Description, SSDD Phase 2 Build 1*, Section 3.7.

Oracle Replication is built into Oracle Serve Enterprise Edition Release 8.1.7 and is thus readily available for use on the US NDC. Oracle Replication has all the means necessary to guarantee reliable replication of any set of processing results tables desired. It can be configured to replicate data in either direction between the PAFB and GAFB sites or even in both directions at once. This capability provides maximum flexibility in formulating recovery strategies for any type of outage that might occur. Oracle Replication can be operated synchronously or asynchronously. Synchronous replication guarantees that every transaction is fully committed on both the master and the target database simultaneously. Asynchronous replication allows batches of transactions to be queued on the master database for deferred execution on the target database. Synchronous replication guarantees maximum moment-to-moment consistency between the master and target, but it would make the performance and availability of the US NDC dependent on the responsiveness and availability of the Alt US NDC and the WAN link. If the WAN link should go down for any reason, updates to the US NDC operational databases would stop until the WAN link was restored. Accordingly, the US NDC application incorporates asynchronous replication between the master and target databases for alphanumeric processing results.

4. Detailed Design of the Database

The detailed design of the US NDC databases is described in terms of four aspects:

- Conceptual Design
- Logical Design
- Internal Design
- Physical Design

In Conceptual Design, a data model of the US NDC System is presented in which the relationships between the data structures contained in the databases are depicted. In Logical Design, the details of the data structures derived from the data model are presented, as well as the organization of these data structures into the system of accounts and schemas upon which the US NDC System processing takes place. In Internal Design, the elements of the database design which are not generally visible to applications are discussed. In Physical Design, the conceptual, logical, and internal design elements are transformed into physical elements which are mapped onto the physical US NDC System architecture. For editorial convenience, some of this material is contained in appendixes and referenced within this section.

4.1 Conceptual Design

This section contains entity relationship diagrams describing the relationships between tables for the US NDC database schema.

The US NDC database schema provides a framework that supports all applications, including real-time and interactive processing, maintenance of a historical data archive and support for seismological research. The database tables are grouped into natural categories:

- Fundamental
- Reference
- Application

Section 4.1.2 describes the Fundamental tables. The Fundamental tables are of general interest and designed to encourage interactive and embedded SQL access by the scientific community. They were made readable and compatible with seismological conventions. The Fundamental tables are dynamic and contain columns used in automated and interactive processing. The information stored in the Fundamental tables includes the following:

- Observed and predicted arrivals
- Events
- Origin hypotheses and the summaries of confidence bounds in origin estimates
- Associations that connect arrivals to origins

- Arrival-based and origin-based measurements of the amplitudes of the seismic signals
- Magnitude estimations of the events
- Descriptive information on the waveforms

Section 4.1.3 describes the Reference tables. These tables are read by many applications but are only updated by the *load_lookup* process. Most of the data in these tables defines the characteristics of seismic and acoustic stations and networks. The information stored in the Reference tables includes the following:

- Station information
- Networks information
- Reference data, such as seismic and geographical regions
- Station-channel information
- Instrument calibration information

Section 4.1.4 describes the Application-specific tables. Unlike the Fundamental and Reference tables, which are of general interest and shared by all applications, the Application-specific tables are used by fewer applications and store application-specific and/or intermediate results. The tables in this category are further grouped by application or subsystem. The applications and subsystems are as follows:

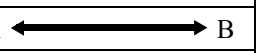
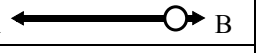
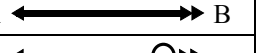
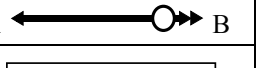
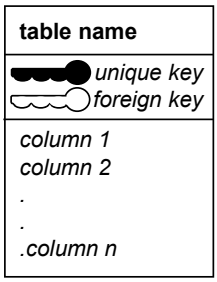
- Interactive Processing
- Map
- Distributed Processing
- Continuous Data Subsystem
- Message Subsystem
- Data Archiving
- Performance Monitoring
- Data Administration
- Event Discrimination

4.1.1 Conventions

The entity-relationship diagrams (ERDs) in this section use the graphical conventions shown in Table 2 to describe relationships, table names, keys, and columns. The table is always shown at the top of the table symbol. Keys, if present, are shown below the table name. The primary key of a table is indicated with a black key symbol (), as is the alternate key. Foreign keys are indicated with a white key symbol (). Keys consisting of multiple columns are shown with a key symbol next to the first column of the key and the remaining columns are listed beneath with no symbol. All column names, if present, are shown below the key section of the diagrams.

Some of the key symbols in the schema have been denormalized for convenience and usability. For readability, columns for denormalized keys are not shown. Furthermore, the primary key, *commid*, of the **remark** table is not explicitly drawn in the tables in which it appears as a foreign key. This section uses the graphical symbols described in Table 2.

Table 2. Entity-relationship Graphical Symbols

DESCRIPTION	SYMBOL
One A maps to one B	
One A maps to zero or one B	
One A maps to many Bs	
One A maps to zero or many Bs	
Database table	

4.1.2 Table Relationships

Relationships between tables are designated by the column or columns through which two tables are related. Table 3 explains the syntax used. In many cases, the column names in the two tables are not identical or a column value in one table must be compared to more than one column value in another table. The delimiters in the syntax are the dash (-) and the slash (/). A dash (-) separates groups of column names from the two tables and a slash (/) separates composite key columns. Other symbols, such as equal (=), ampersand (&), and parentheses ( ) specify how the columns are compared. Expressions within parentheses in relationships between tables are evaluated first; the order of operations is = and &.

Table 3. Syntax Used to Indicate Database Table Relationships

SYNTAX	DEFINITION
col	This is the simplest case where the column names (*col*) of the keys are the same at each end of the relationship. Both keys consist of a single column
col1/*col2*	A slash (/) is used when a key is comprised of multiple columns. Here, the keys in both tables are the same and consist of two columns, *col1* and *col2*
col1-col2	A dash (-) is used when the column names of the keys in the two tables are not the same. Col1 is the name of the key column in one table and col2 is the name of the key column in the other. Each key consists of a single column. Only one dash may be used and the dash separates the keys of the two tables. A dash can be combined with a slash (/) to show that the keys consist of multiple columns and that one or more of the columns have different names in the two tables, as in *col1/col2-col3/col4* (both parts of the key are different in the two tables) or *col1/col2-col1/col3* (only the second part of the key is different in the two tables)
col1-col2/*col3*=`value`	An equal sign (=) is used when a component of a key must be set to a particular value. Here *col1* is the name of the key column in one column. *Col2* and col*3* must be set to the shown value. See the *arid-tagid*/*tagname*=`arid` relationship between **arrival** and **wftag** and *orid*/*tagid*/*tagname*=`orid` relationship between **origin** and **wftag**
col1-col2&*col3*	An ampersand (&) is used to show that a key in one table may have a value between the values of two keys in another table. Here the value of *col1* must be between the values of *col2* and *col3*. See the *sta*/*chan*/*time-sta*/*chan*/*time*&*endtime* relationship between **wfdisc** and **sensor**
(*col1*)-(*col2*)	Parentheses () are used to show that the keys within them have different formats and a conversion must be made to make the comparison. Here *col1* corresponds to *col2* but *col1* and *col2* have different storage formats (usually an epoch time versus a date). See the *sta*/*chan*/(*time*)-*sta*/*chan*/(*ondate*&*offdate*) relationship between **sitechan** and **siteaux**

The example shown in Figure 2 with **table_1** on the left and **table_2** on the right, demonstrates the possible relationships between the two tables. The syntax defined in Table 3 is used to interpret the relationships between the figure's tables. *Col8* in **table_2** has no matching column in **table_1** and must be equal to value in this relationship. All other columns have one or more corresponding columns in the other table. Following the syntax, *col1* in **table_1** must have the same value as *col1* in **table_2** and *col2* in **table_1** must have a value between *col4* and *col5* in **table_2** for the one-to-many relationship indicated by the entity-relationship symbol to be true.

Figure 2. Sample Entity-relationship

Some of the entity-relationship diagrams show multiple relationships between two tables. For example, there are two relationships between the **origin** and **event** tables in Figure 2; a many-to-zero or many-to-one relationship through *evid* and a zero-to-one or one-to-one relationship through *prefor-orid*. The *evid* relationship states that for every **origin** entry, there is zero or one corresponding entry in **event** where the *evid* in **origin** equals the *evid* in **event**, and for every event entry, there are many **origin** entries where the *evid* in **event** equals the *evid* in **origin**. The *prefor-orid* relationship states that for every **origin** entry, there is zero or one corresponding entry in **event** where the *orid* in **origin** equals the *prefor* in **event**, and for every **event** entry, there is one **origin** entry where the *prefor* in **event** equals the *orid* in **origin**.

4.1.3 Fundamental Tables

Figure 3 shows the summary of the Fundamental tables and keys. Each of these tables is involved in preserving origin hypotheses and events. The subset of the Fundamental and Reference tables based on the CSS Version 3 scheme are designated as css 3.0 in the following figures. The tables for the new representation of amplitudes and magnitudes are designated as css 3.1.

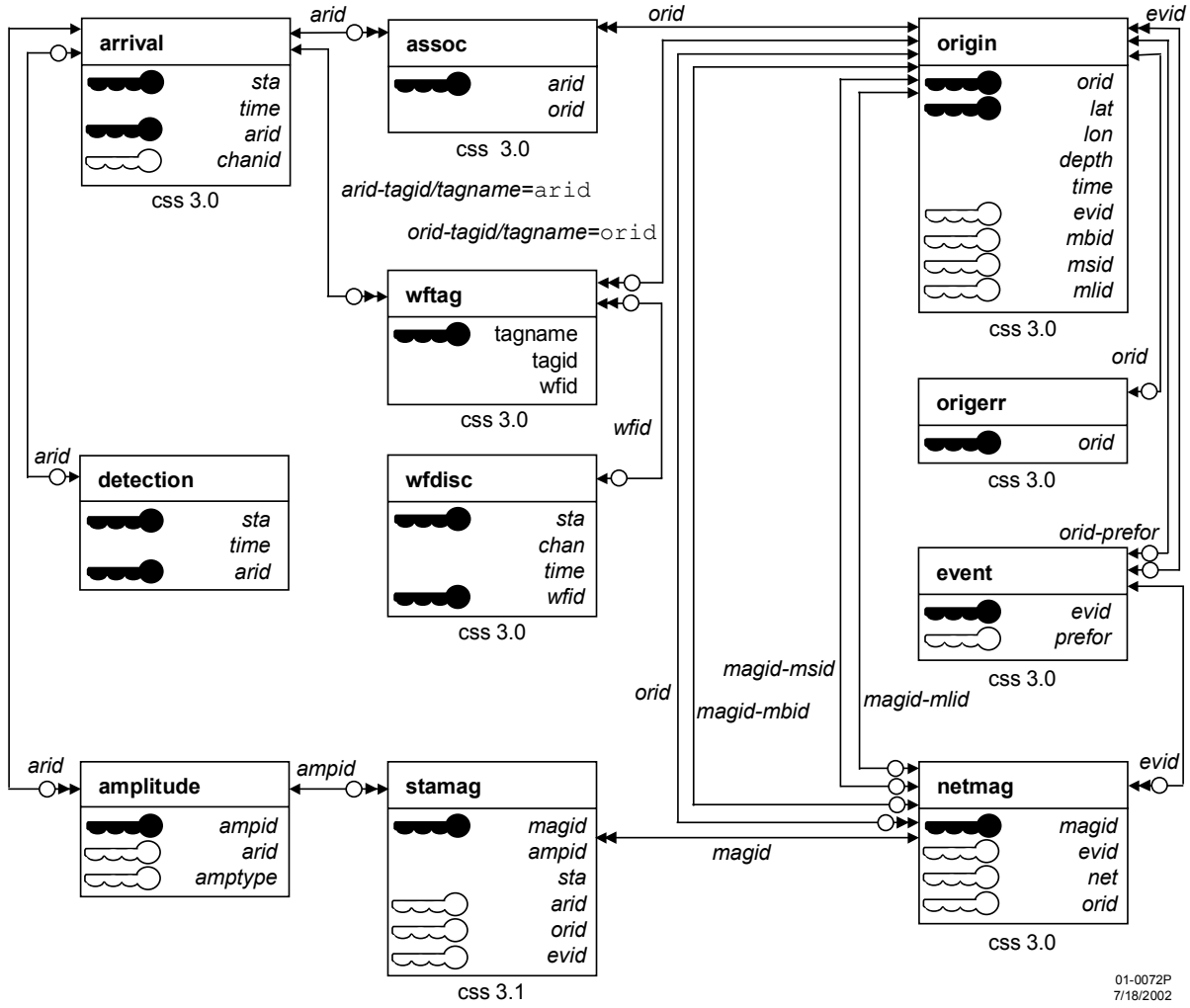


Figure 3. Relationships between Fundamental Tables

4.1.3.1 Events

Figure 4 shows tables involved in preserving origin hypotheses and events. During automated processing, arrivals in the **arrival** table are associated with different origin hypotheses from the **origin** table through the **assoc** table. Based on the seismic signal measurements, groups of arrivals are associated with presumed events. Each event may have up to three different origin hypotheses, each with a different event location estimate. The preferred origin hypothesis is specified in the **event** table as a *prefor*.

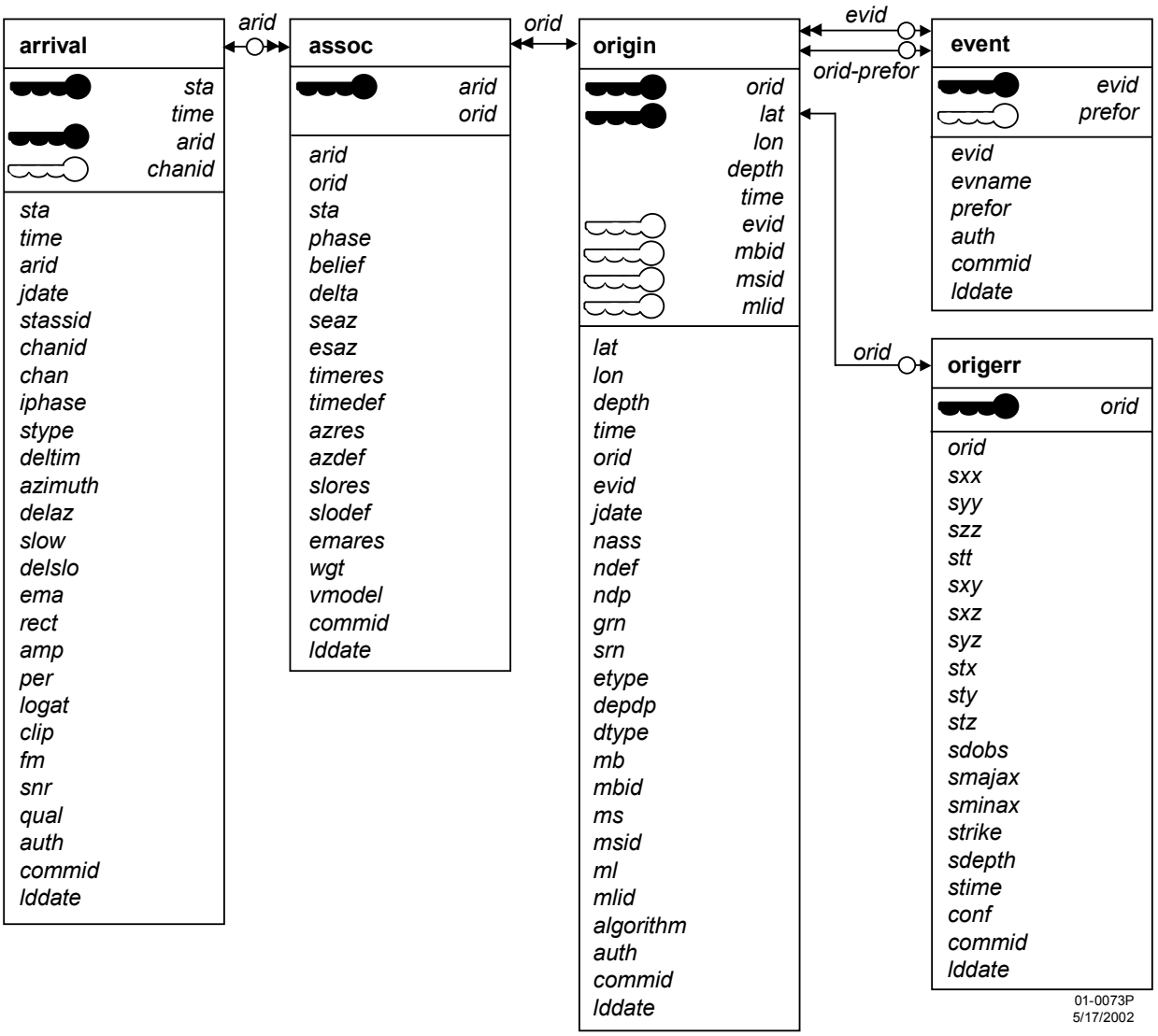


Figure 4. Event/Arrival Table Relationships

4.1.3.2 Arrival-centric Data

Figure 5 shows the tables that contain arrival-centric data on a specific detection made with different methods. Amplitude measurements made on three-component data come from the **amp3c** table. The **apma** table contains results of a particle motion analysis. The **hydro_arrival** table provides hydroacoustic arrival information. The **detection** table keeps the summary information about seismic and hydro detections. The **hydro_assoc** and **hydro_arr_group** tables provide storage for hydro azimuth estimates. The relationships between the **arrival** table and the arrival-centric tables through an *arid* is one to zero or one (or one to zero or many for the **amp3c** table).

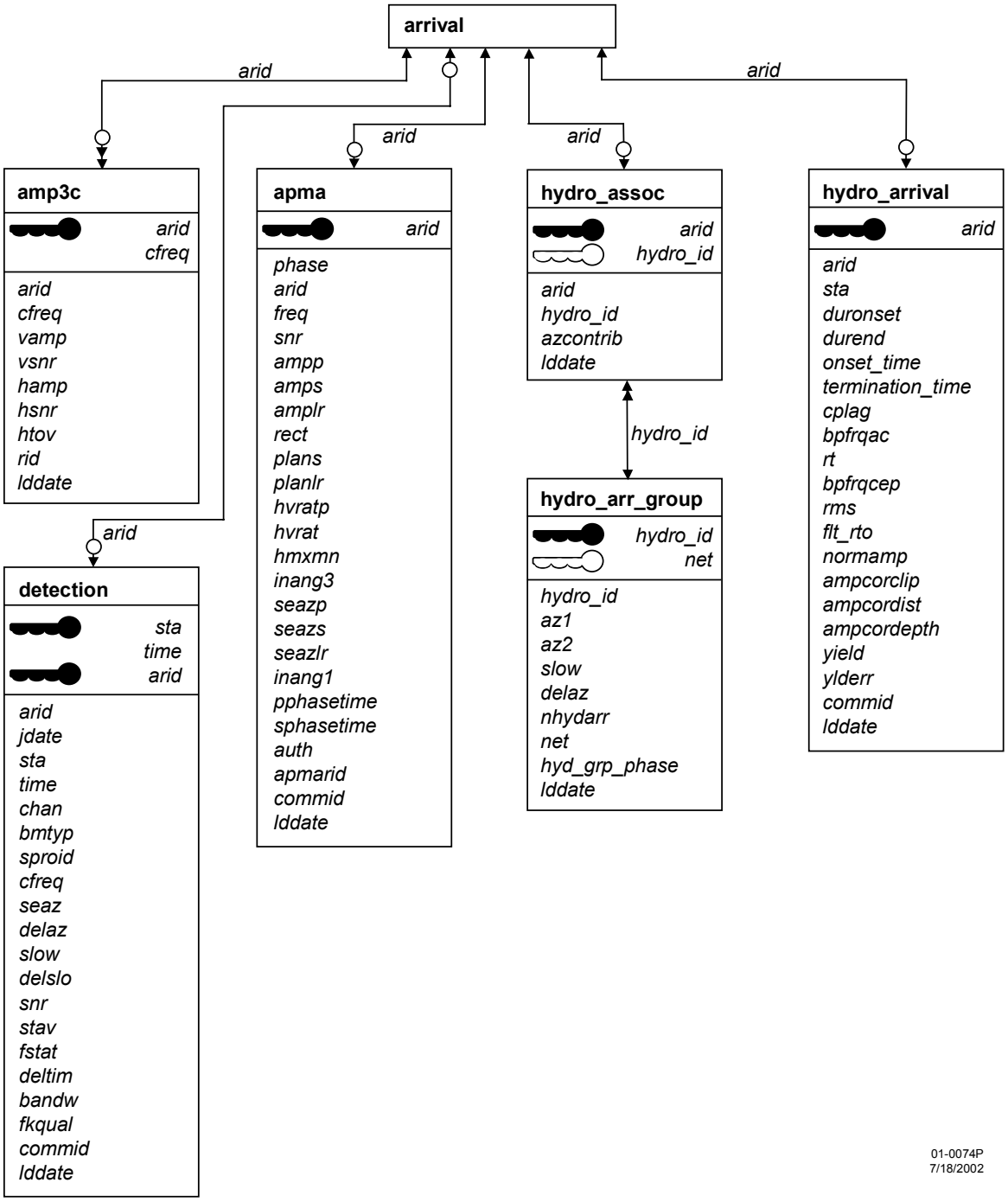


Figure 5. Arrival-centric Table Relationships

4.1.3.3 Amplitude/Magnitude Measurements

Figure 6 shows tables that contain seismic signal amplitudes (**amplitude** table) and magnitude estimations for stations and events (**stamag** and **netmag** tables). The **stamag** table performs a

function similar to the **assoc** table in that it links the **amplitude** and **netmag** tables. The **stamag** table also provides the capability to flag magnitudes as defining or not defining through the *magdef* column. The **ampdescript** table contains descriptions of how amplitude measurements were made and is related to the **amplitude** table through the amplitude measure descriptor *amptype*.

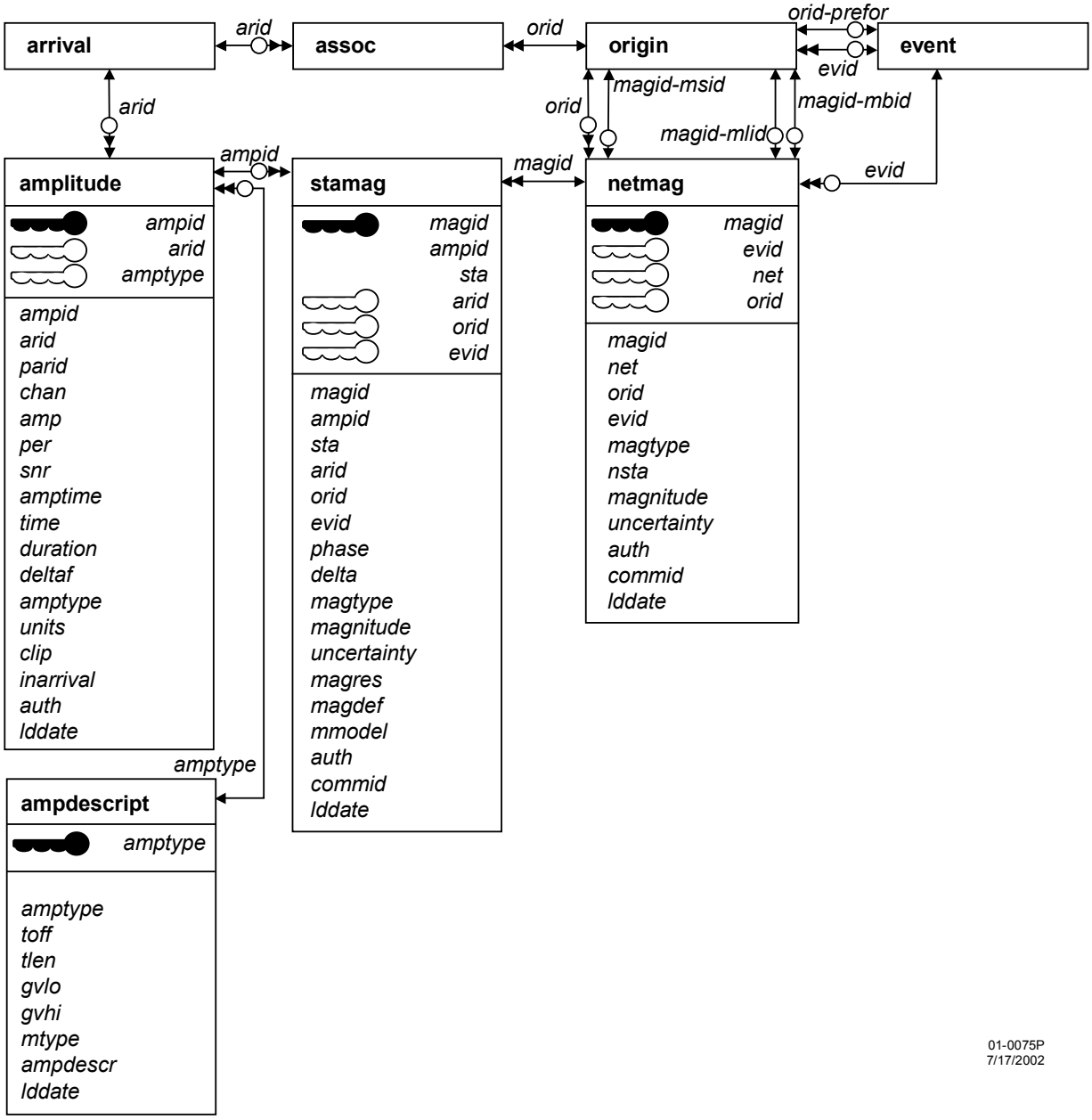


Figure 6. Amplitude/Magnitude Table Relationships

4.1.3.4 Waveforms

Figure 7 shows the Waveform Tables **wfdisc** and **wftag**. The waveforms themselves are stored in the flat files on the disk. They are usually called ".w" files and are a sequence of a sample values (usually in a binary representation). The descriptive information on the waveforms is stored in the **wfdisc** table, which provides a pointer (or index) to the waveforms on the disk. The **wfdisc** table is linked to the **arrival** and **origin** tables through *sta*, *chan*, and *time*. The **wftag** table specifies which table the **wfdisc** record is linked to, **origin** or **arrival**.

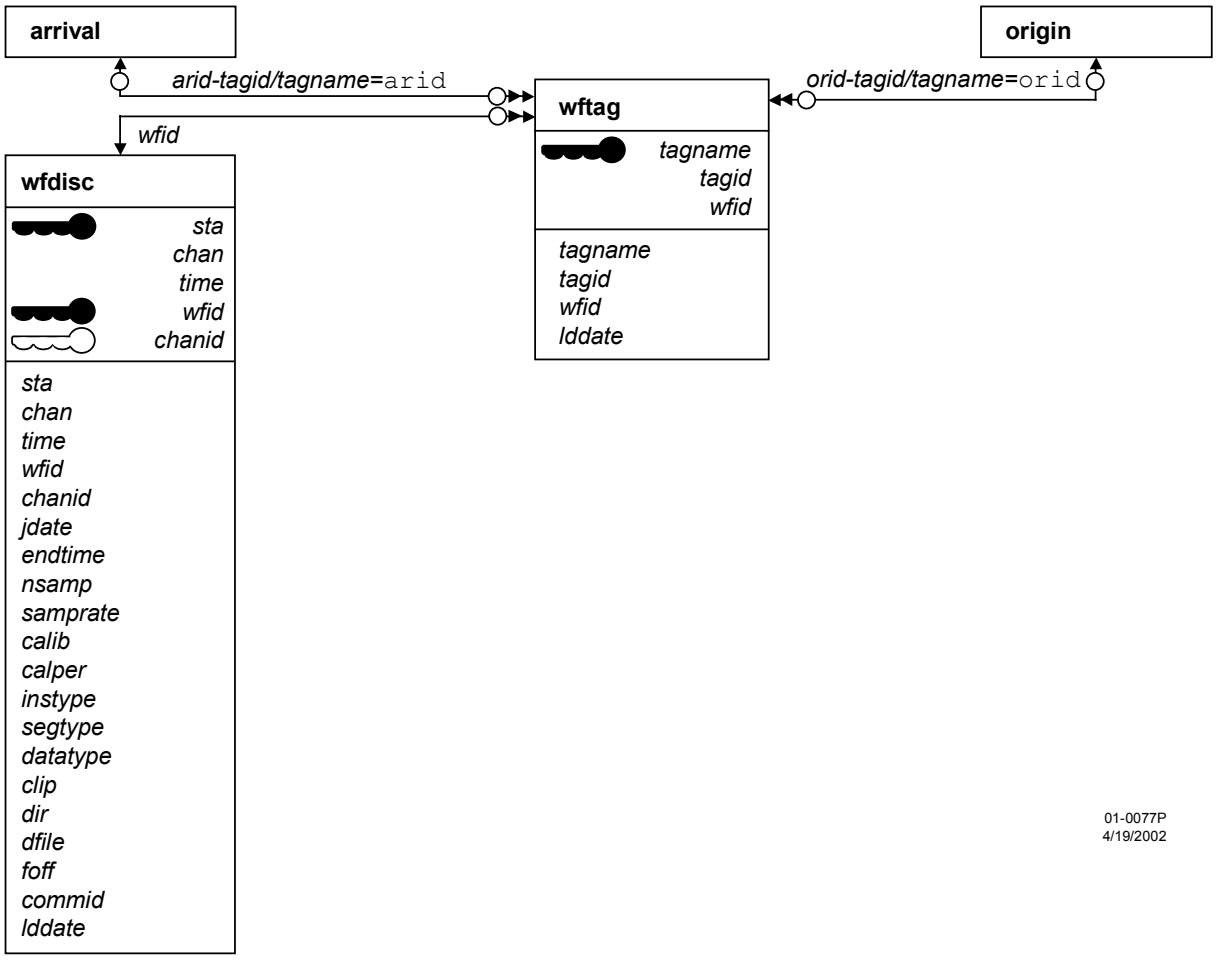


Figure 7. Waveform Table Relationships

4.1.4 Reference Tables

The Reference tables are fairly static and primarily contain look-up information. Figure 8 is an overview of the tables in this category. Figures 9 and 10 describe these tables in detail.

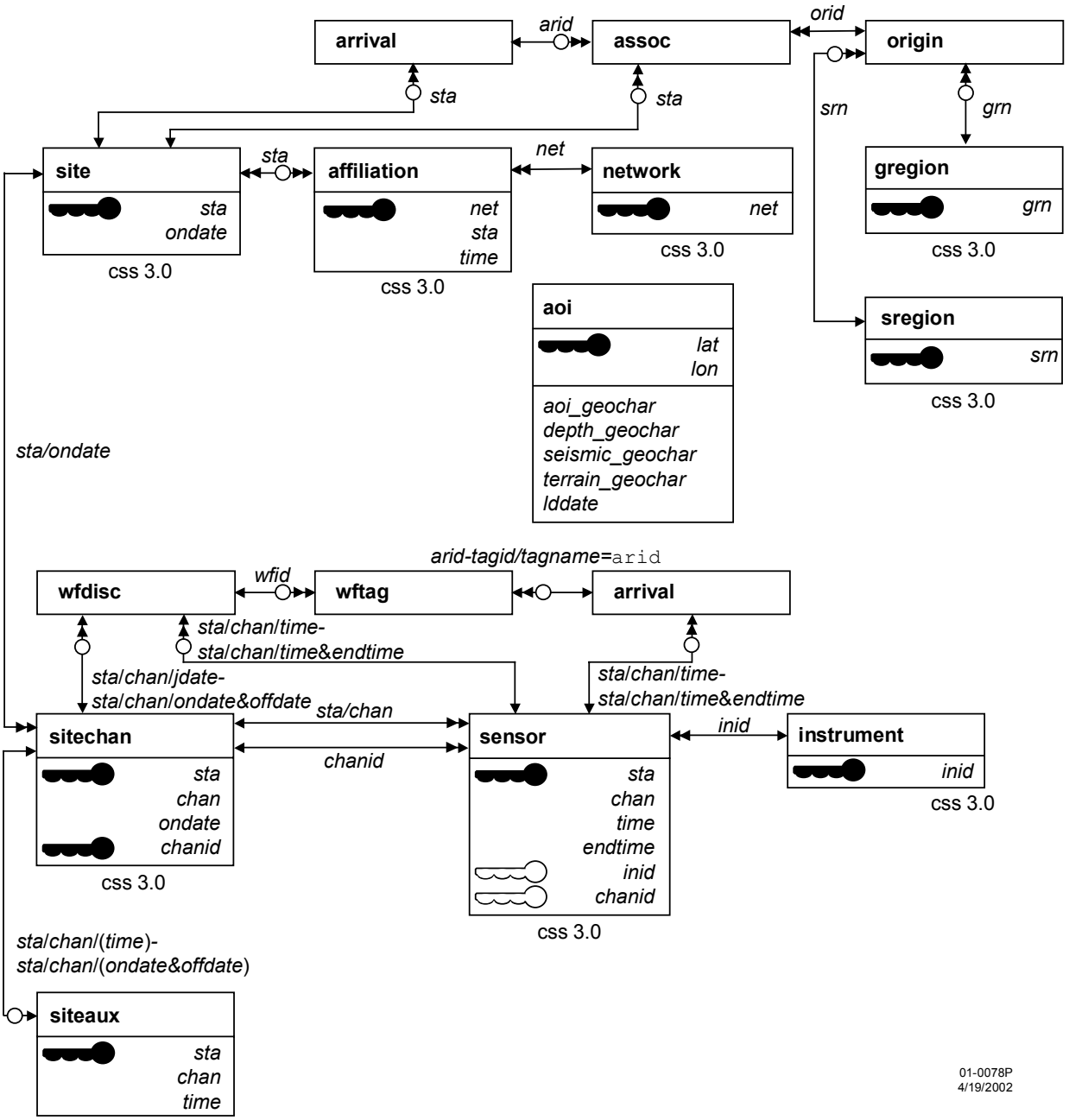


Figure 8. Reference Table Relationships

4.1.4.1 Network Information

Figure 9 shows tables related to networks. The **site** table contains station location information. It describes the geographic location of a station. The **site** table also contains fields that describe the offset of a station relative to an array reference location. The **affiliation** table groups stations

across wide geographic areas as networks. The general information about the seismic networks is stored in the **network** table. The **sregion** table contains seismic region numbers and their descriptions. The **gregion** table contains geographic region numbers and their description. The **gregion** and **sregion** tables are related to the **origin** table through geographic region number *grn* and seismic region number *srn*, correspondingly.

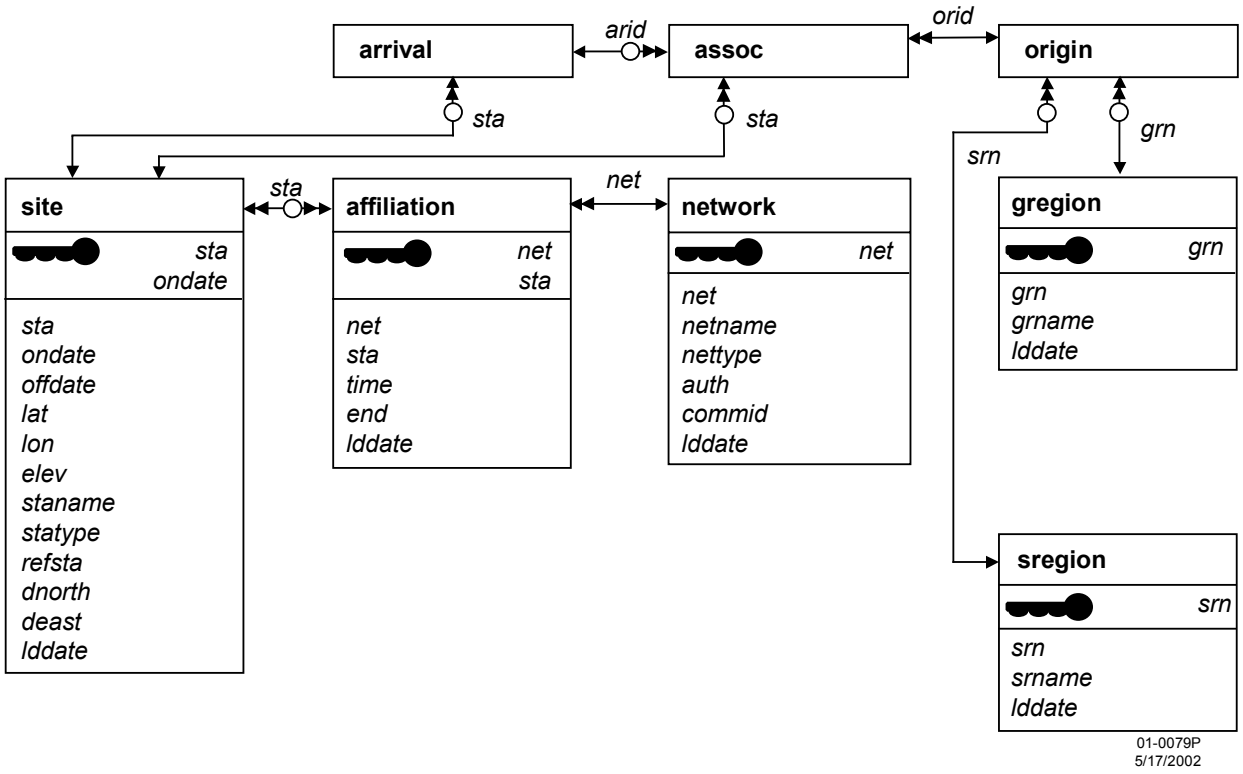


Figure 9. Network Table Relationships

4.1.4.2 Channel Information

Figure 10 shows tables that contain specific information about station channels. The **siteaux** table holds additional information that is not included in the **site** table. It is linked to the **site** table through the **sitechan** table, which contains station-channel information. Detailed calibration information is stored in flat files, in a variety of formats. The **instrument** table holds complete instrument response information, including ancillary calibration information and pointers to the flat files with detailed instrument responses. The instrument identifier *inid* links the **instrument** table to the **sensor** table. The **sensor** table contains calibration information for specific sensor channels and is linked to the **wfdisc** and **arrival** tables through *sta*/*chan*/*time*. It provides instrument update records, using the calibration period column *calper*, thus linking a *sta*/*chan*/*time* to a complete instrument response.

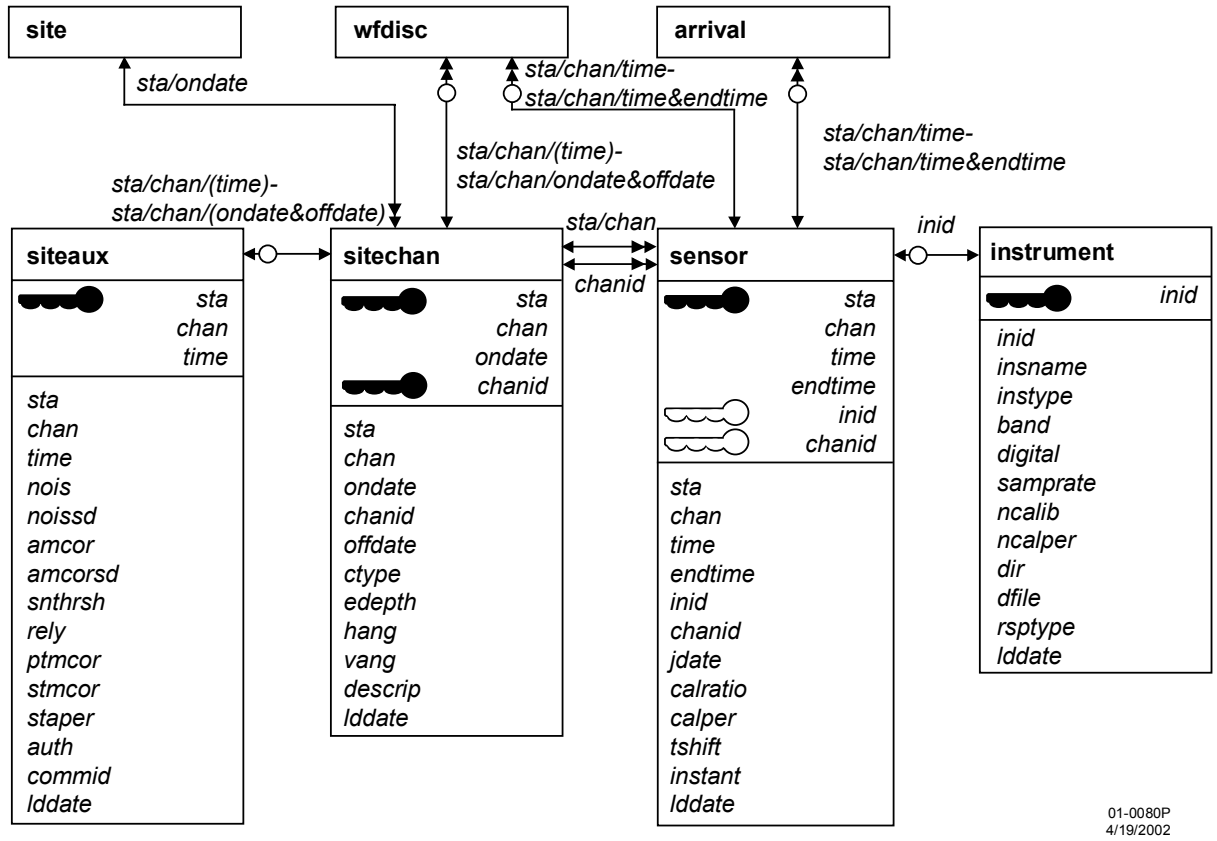


Figure 10. Channel Table Relationships

4.1.5 Application-specific Tables

This section contains tables that are specific to an application or subsystem.

4.1.5.1 Interactive Processing

Figure 11 presents application-specific tables that are used in Interactive Processing applications. The **event_control** table preserves the specific user-defined controls that were used to determine the location and magnitude of a given *orid*. The location and magnitude programs preserve the values of event location and magnitude key parameters that analysts set while reviewing the automatic processing results from the **event_control** table. This table is also used in post-analysis processing. The **remark** table contains freeform comments for many tables in the schema. The **ga_tag** table contains information on the use of arrivals and origins in the *GA* application. The tables in Figure 11 are also used by event quality control software.

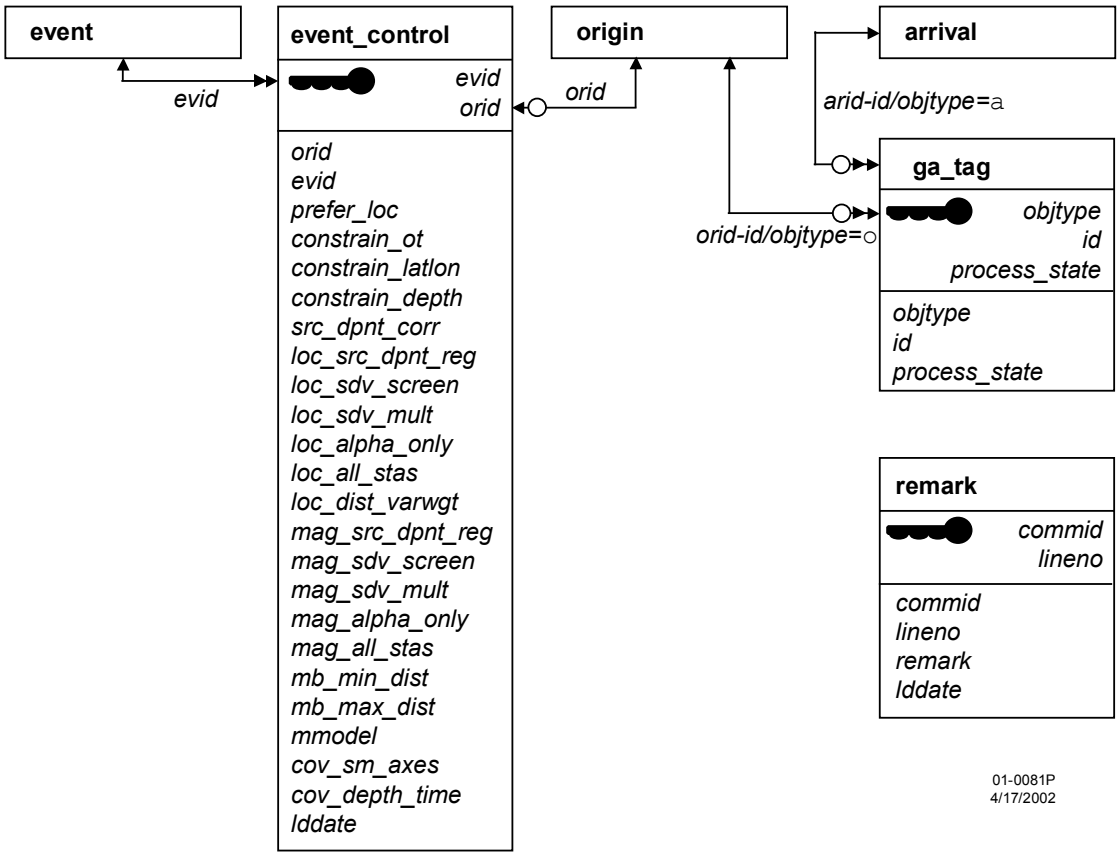


Figure 11. Relationships of Tables Used in Interactive Processing

4.1.5.2 Map

Figure 12 shows the tables for the *Map* application. The tables in this group are not linked to any of the core tables. The **mapdisc** table stores information about maps that are on the disc. The **colordisc** and **mapcolor** tables allow plotting the same map in different colors. The **overlaydisc** and **mapover** tables contain the information on the maps' overlays. The **mappoint** table stores the labeled point data to be displayed by the *Map* application.

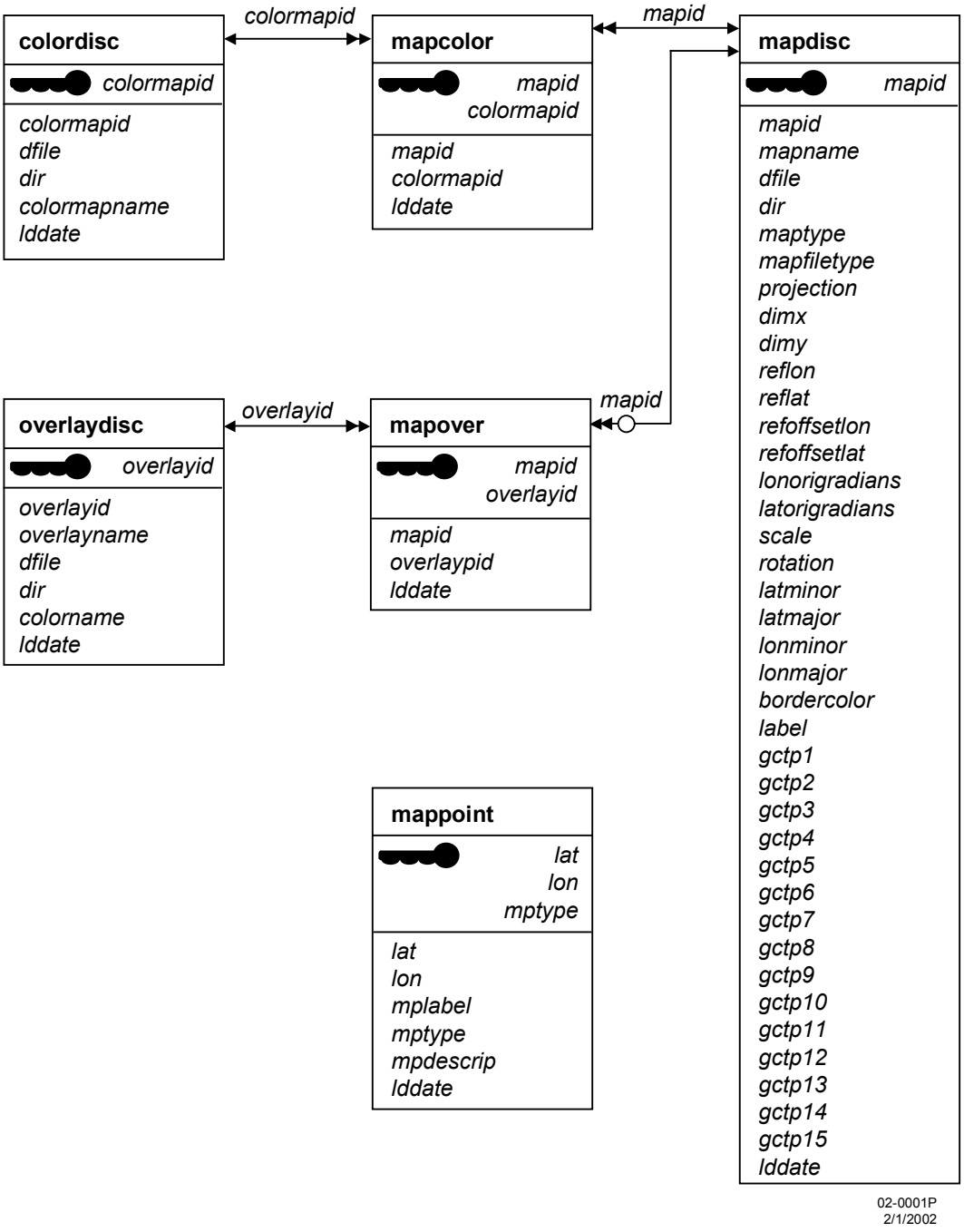


Figure 12. Map Table Relationships

4.1.5.3 Distributed Processing

Figure 13 shows how the Distributed Processing Subsystem manages jobs and orchestrates the workflow. The **interval** table has the columns *time*, *endtime*, and *name* that define starting time, ending time, and the name of the processing time interval for a named object. The *class* column allows the **interval** table to be used for different classes of objects. The **timestamp** table is used for scheduling automatic processing of time-series data. The **interval** and **timestamp** tables are also used by some of the Data Services applications. The **stanet** table is a clone of the **affiliation** table; it contains stations for array mapping and groups array sites into an array network for Distributed Processing.

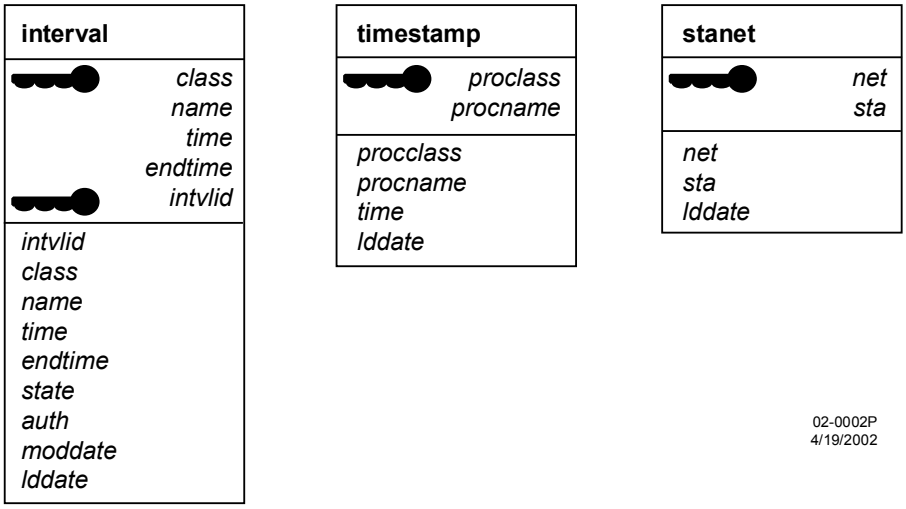


Figure 13. Tables Used by Distributed Processing Applications

4.1.5.4 Continuous Data Subsystem

Figure 14 shows relationships between tables of the Continuous Data Subsystem. The **dlman** table holds the data about currently running *DLMan* instances. The **wfconv** table contains information about the computers on which incoming data is stored for processing, and the table **alphasite** contains information about the computers to which data is forwarded. Administrative information for the Continuous Data Subsystem is stored in the **dlfile** table. The tables **calibrate** and **channame** contain the supportive information for data processing.

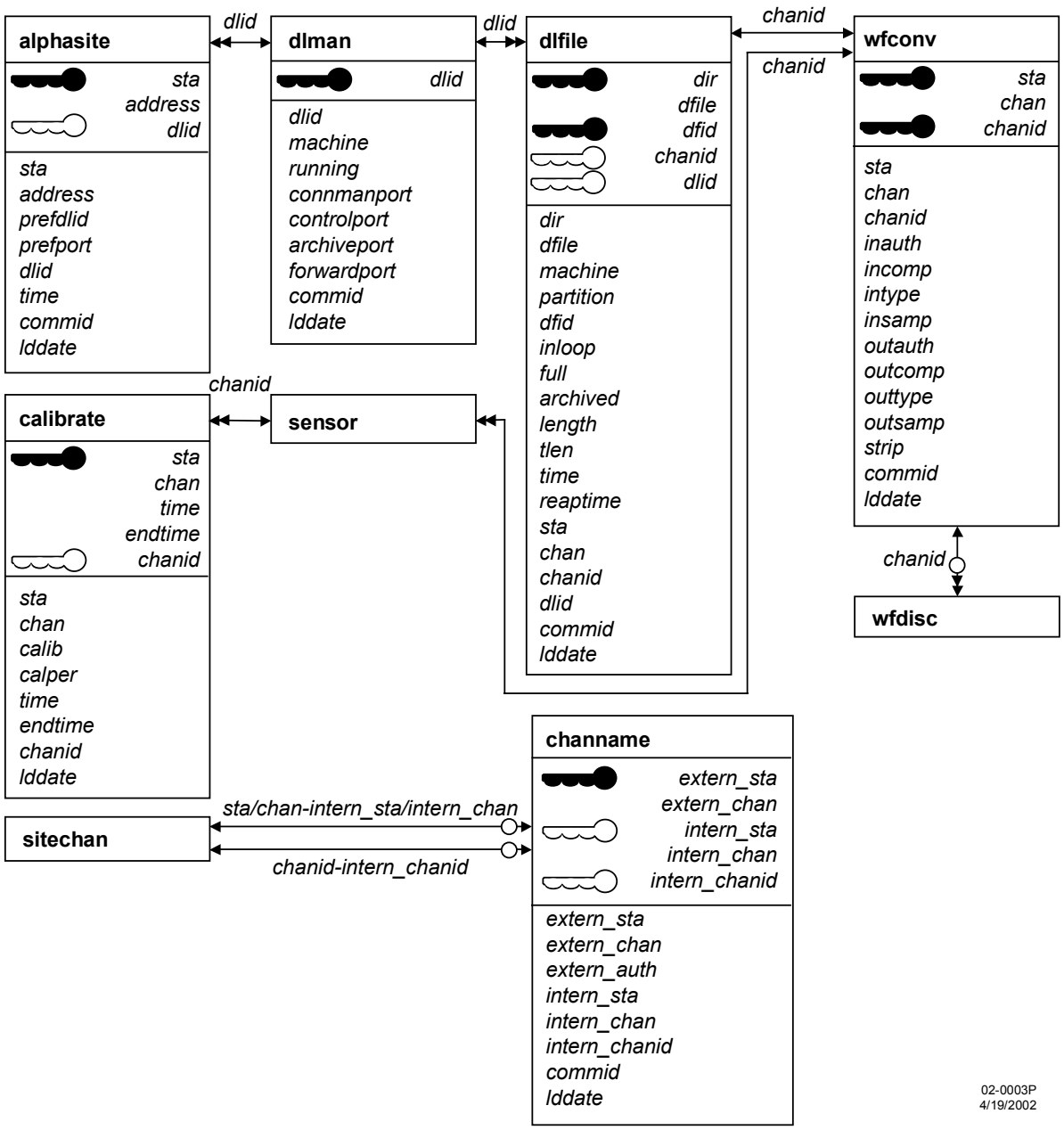


Figure 14. Continuous Data Subsystem Table Relationships

4.1.5.5 Message Subsystem

Figure 15 shows relationships between tables for the Message Subsystem. The **datauser** table tracks authorized users of the Message Subsystem. The **request** table defines segments of auxiliary waveform data to be acquired. Data import programs must succeed in acquiring all the data for a time interval before changing the state to indicate success. The **msgdisc**, **msgdest**, **msgdatatype**, and **msgaux** tables contain information about messages. The **msgdisc** table

information includes the date and time that the message was sent or received, identification information, and where the message is stored. The **msgdatatype** table supports data tracking by recording each data section in a message for incoming and outgoing data messages. The **msgaux** table contains records of unsuccessfully processed messages. The **ftplogin** and **ftpfailed** tables are used by the auxiliary data retrieval system to obtain data via ftp from auxiliary stations.

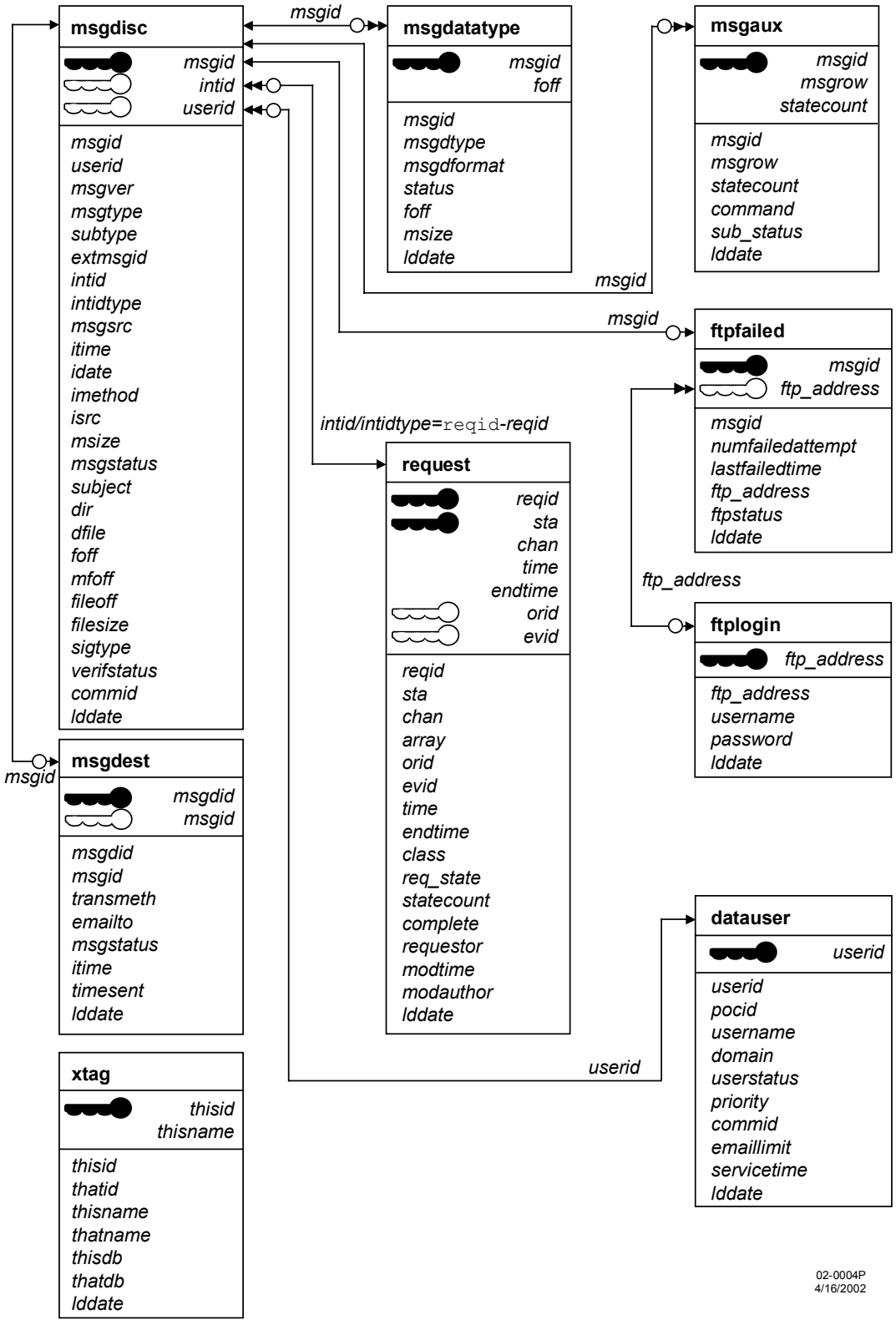


Figure 15. Message Subsystem Table Relationships

4.1.5.6 Data Archiving

Figure 16 shows the tables of the Data Archiving Subsystem, which contain information used by the software subsystems that migrate database tables between databases and that migrate timeseries data to the mass-storage device. The **mig_rules** table contains rules for migrating database tables from one database table to another. The **mig_date** table is used to track table migration. The **wfactivity** table explains the descriptive information on the waveforms in the wfdisc table for a channel group and time region. The **wfaudit** table tracks the audit changes in the **wfdisc** table. The **chan_groups** table indicates which *sta*/*chan* pairs belong to a given *class*/*name* (**wfactivity**) group. The **interval_files** table provides the administrative information for the archived interval files.

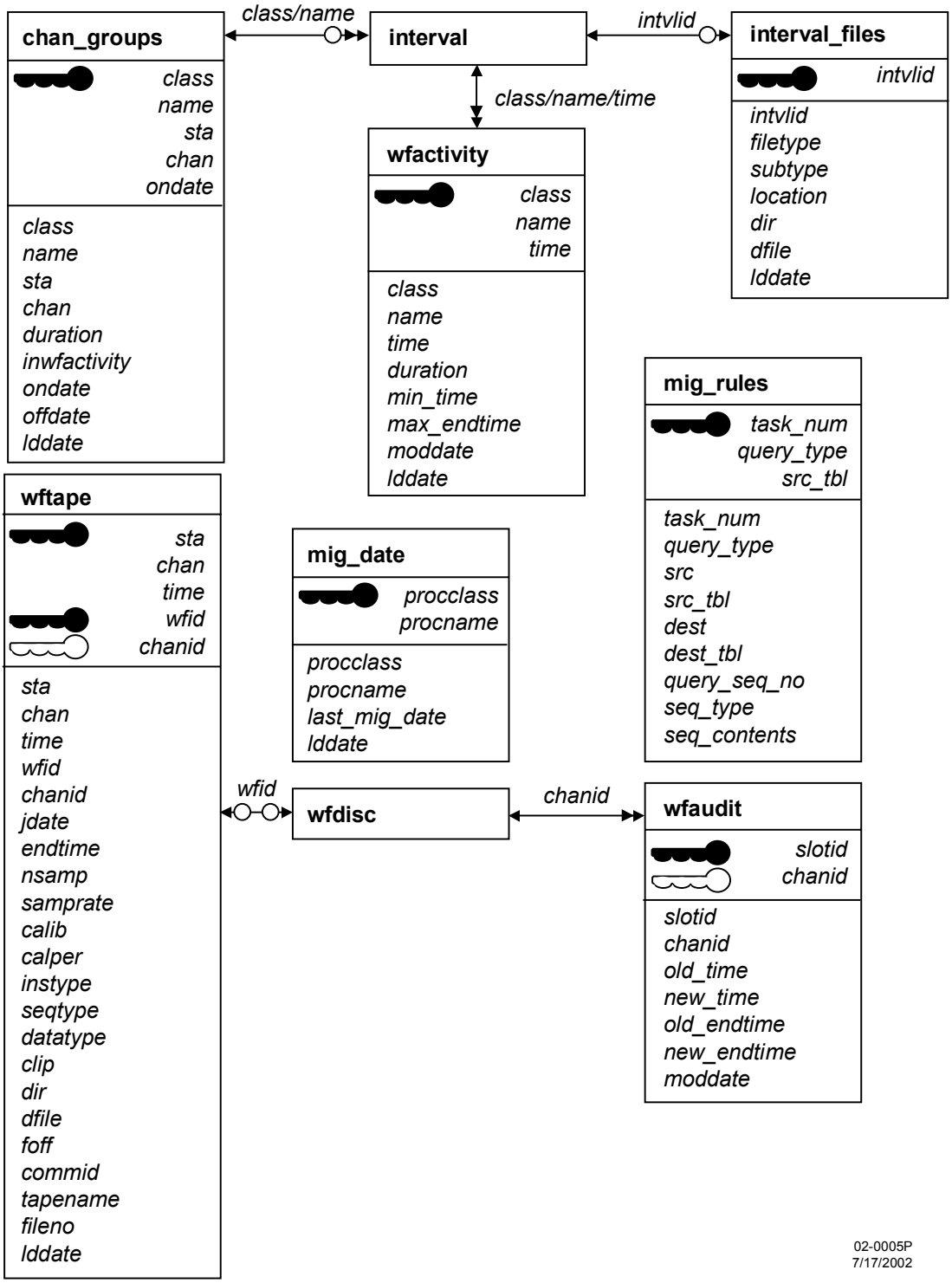


Figure 16. Data Archiving Subsystem Tables

4.1.5.7 Performance Monitoring

Figure 17 shows Performance Monitoring tables that hold data pertinent to monitoring scientific performance. The **bull_comp** table contains results from the *BullComp* application on the comparison of two seismic bulletins. The **ev_summary** table contains statistical summary analysis of automated and analyst solutions from the *ExAnComp* application. The **ex_an** table contains the comparison of the analyses of automated solutions against the analyst solutions from the *ExAnComp* application. The **qcstats** table contains waveform data quality statistics. Both **qcdata** and **qcstats** are populated by *DFX*. The **qcdata** table contains performance monitoring data quality information. The **missed_class** table contains information pertaining to events identified by only one bulletin during a bulletin comparison.

Figure 18 displays the tables that hold additional station data for Performance Monitoring. The **station_hist** table contains the station's processing history. The **station_type** table keeps the station type information. The **datadays** table stores the days and times for which data is available for Performance Monitoring. The **datacollected** table records information for Performance Monitoring to determine if image generation can be performed. The **pixdisc** table records the generated images.

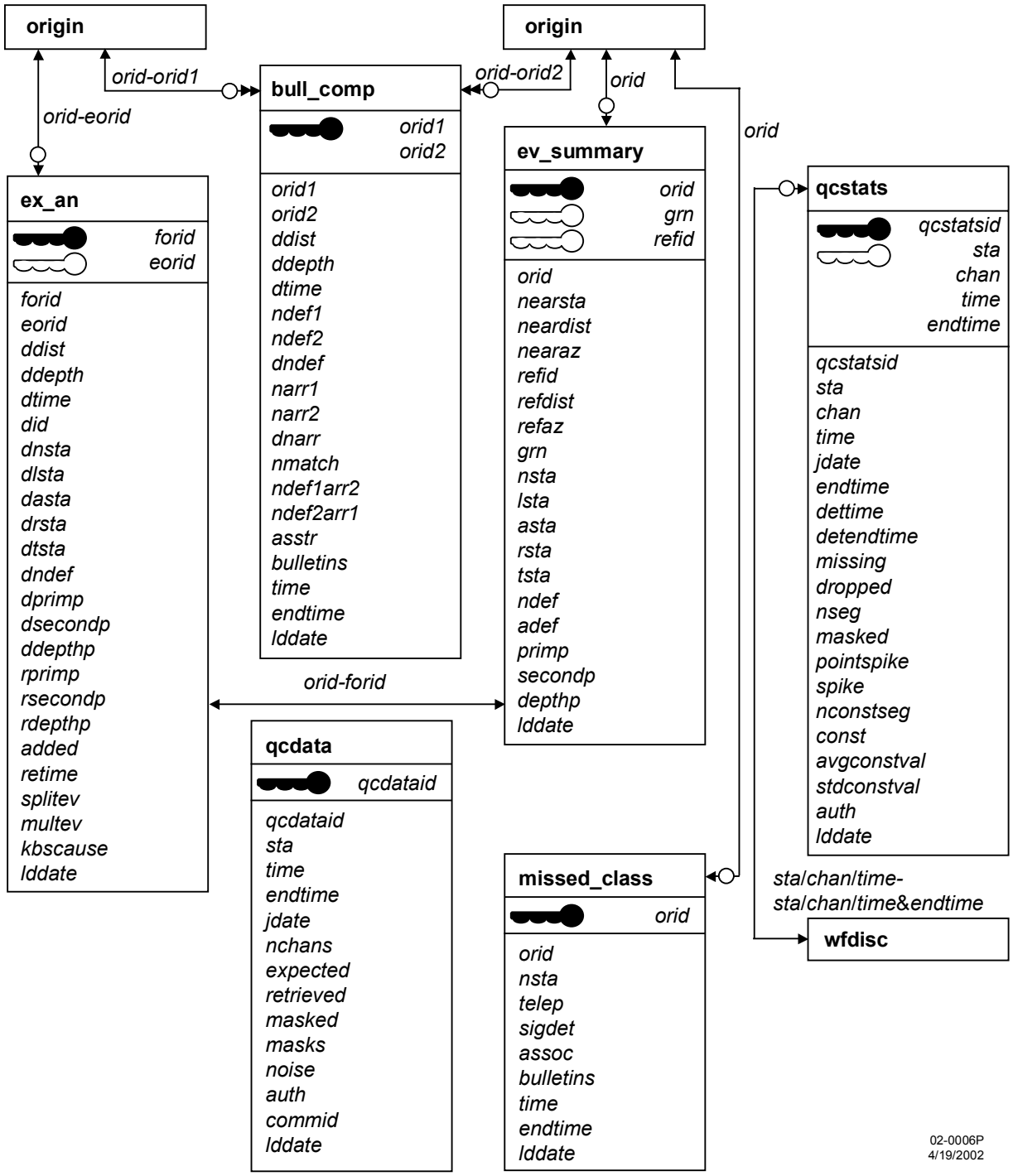


Figure 17. Performance Monitoring Table Relationships

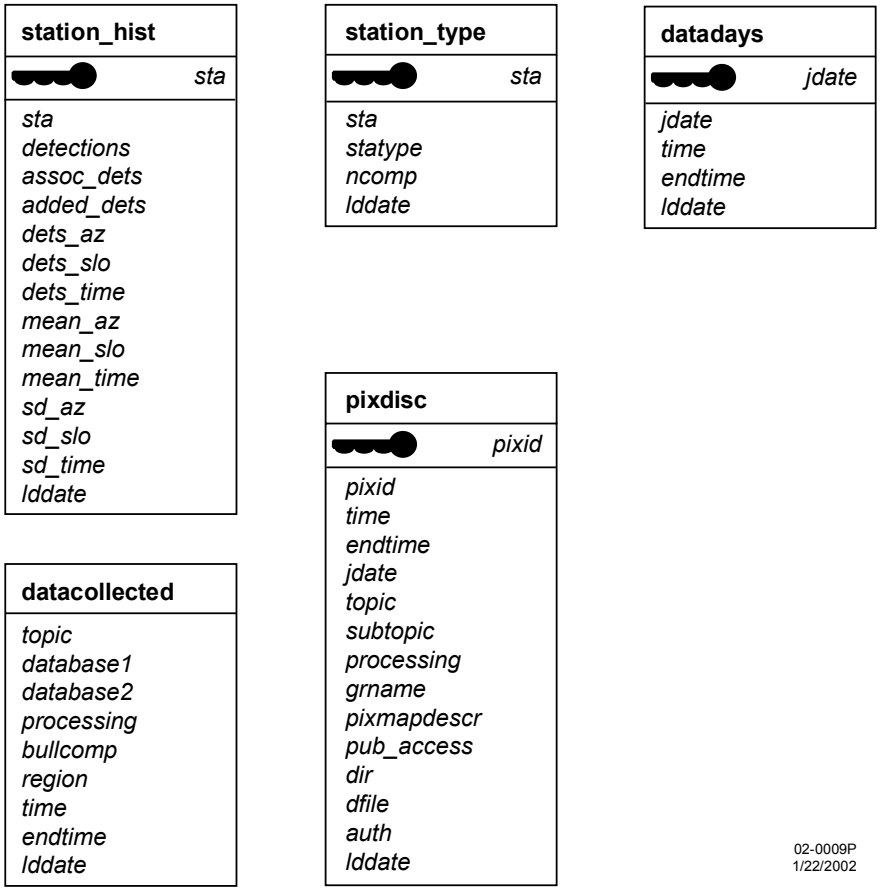


Figure 18. Additional Performance Monitoring Tables

4.1.5.8 Identifier Management

The **lastid** table (see Figure 19) facilitates the use of the database. The **lastid** table contains the last value used for the numeric keys or identifiers. Programs retrieve new sequential keys values by requesting the last key used and incrementing the value in the table by the number of key values requested.

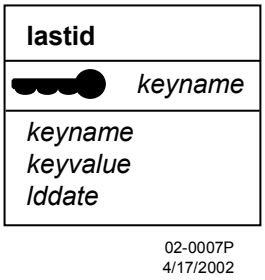


Figure 19. Tables Used for Identifier Management

4.1.5.9 Event Discrimination

Figure 20 shows tables involved in the event discrimination process. The **discrimvote** and **discrimuse** tables are related to the **origin** table through *orid*. The **hydro_origin** table contains hydroacoustic origin information and is related to the **origin** table through *orid*.

The **discrimuse** table contains information on the use/nonuse of station data in discriminant voting. It identifies, for each station associated to the **origin** table, the use or non-use of that station's data in the discriminant vote for six different discriminants. The **discrimvote** table identifies the vote value for each of the discriminants used in the event classification. The **hydro_origin** table contains a summary of AFTAC-specific hydroacoustic origin information. This table also identifies if this origin is part of a series and the unique identifier for that series.

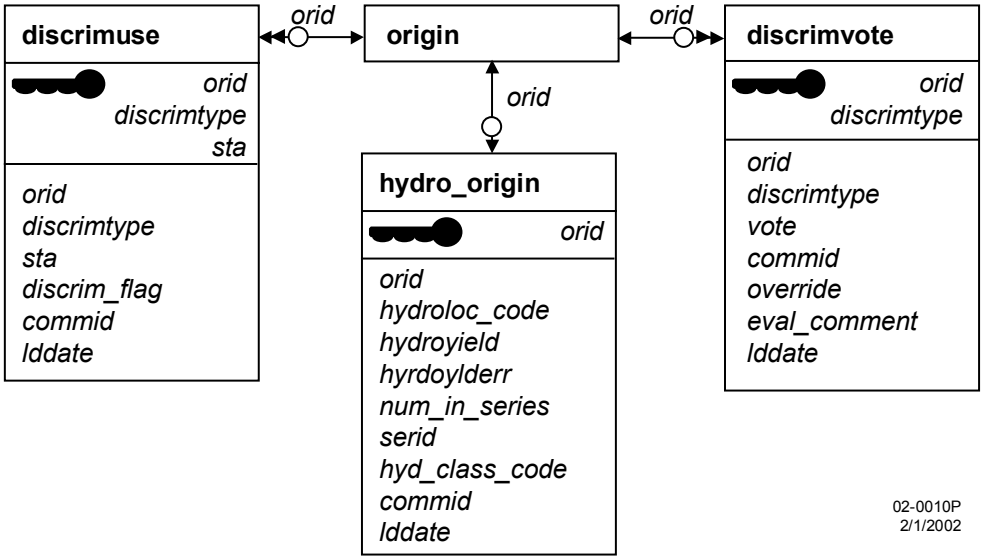


Figure 20. Event Discrimination Table Relationships

4.2 Logical Design

Two of the most influential factors driving US NDC database design are the requirements to support data acquisition and pipeline processing. The US NDC database accounts support one or more pipelines and some of the accounts support both data acquisition and pipeline processing. The pipeline accounts are designed to store results from a particular stage of processing and make those results available to subsequent stages of processing through the use of access permissions and private and public synonyms as described in Section 3.5.1.

4.2.1 Overview of Data Acquisition and Pipeline Processing

Data acquisition is accomplished by a series of processes that parse, manipulate, forward, and store incoming data. Depending on data format, one of three processes writes the data to an intermediate storage area on the UNIX filesystem called the framestore. From the framestore, the data is parsed and written to the diskloops, or forwarded to the Unclassified Archiving Subsystem and the Classified System or to outside customers.

Once the data is received by the Classified Acquisition Subsystem (CAS), it is processed according to data type and processing goals. Data processing is accomplished in four pipelines. Each pipeline is characterized by a set of processes and the database accounts, which are read from and written to by those processes. The main purpose of each pipeline is summarized below:

- The Global pipeline builds a global bulletin with short period (SP) and long period (LP) seismic data and performs classification for events of interest using seismic data and hydroacoustic arrivals associated with offshore events. It also uses seismic data recorded at selected stations located at regional distances from the regions of interest [formally a Broad Area Regional Monitoring (BARM) pipeline functionality].
- The Spotlight pipeline builds a regional bulletin for specified areas and estimates regional magnitude.
- The Forward pipeline provides rapid notification of events in configurable-tunable, pre-defined target areas and validates and refines target events on an accelerated schedule, making the data available for rapid analysis.
- The Hydroacoustic pipeline detects and identifies hydroacoustic phases with high signal- to-noise ratio.

Each pipeline performs a specific set of processes. Scheduling of the processing is initiated, based on data availability, employing an intelligent algorithm. The algorithm strives for completeness, but waits only a limited time for missing data. Data that arrive too late for a processing stage are subjected to preliminary processing and inserted into a later processing stage, as feasible. Of the four pipelines, the Global pipeline is the most complex and extensive processing model. See *SAIC-02/3009*, *SSDD Phase 2 Build 1 (US NDC)*, for a more detailed description of data acquisition and pipeline processing.

4.2.2 Accounts and Schemas

As described in the previous paragraphs, the US NDC database account structure was designed to support two primary functions: Data acquisition and data processing. The pipeline processing and data acquisition accounts cannot meet these goals without additional system, maintenance, archiving, and special purpose accounts. For convenience and readability, only the data acquisition and pipeline accounts are listed in the tables below. For information on the system, maintenance and special purpose accounts, see the description in Sections 4.2.2.3 through 4.2.2.5. Tables 4 and 5 list the database accounts and the function(s) that they support.

Appendix D indicates which objects from Appendixes A and C are incorporated into the schema associated with each US NDC database account.

Table 4. Unclassified Database Accounts and Functions

DATABASE NAME	ACCOUNT	FUNCTION SUPPORTED
OPSDB	GLOBAL	Data acquisition
OPSDB	LOOKUP	Data acquisition
ARCHDB	GLOBAL	Data archive
ARCHDB	LOOKUP	Data archive

Table 5. Classified Database Accounts and Functions

DATABASE NAME	ACCOUNT	FUNCTION SUPPORTED
OPSDB	AL1	Global pipeline
OPSDB	AL2	Global pipeline
OPSDB	DETPRO	Global pipeline
OPSDB	EVAL1	Global pipeline
OPSDB	EVAL2	Global pipeline
OPSDB	FAL	Look-forward pipeline
OPSDB	GLOBAL	Data acquisition and all pipelines
OPSDB	HAL	Hydroacoustic pipeline
OPSDB	HYDRODET	Hydroacoustic pipeline
OPSDB	LFDET	Look forward pipeline
OPSDB	LOOKBACK	Lookback processing
OPSDB	LOOKUP	Data acquisition and all pipelines
OPSDB	MIGRATE	Data migration
OPSDB	MONITOR	Performance monitoring
OPSDB	RAL1	Spotlight pipeline
OPSDB	RAL2	Spotlight pipeline
OPSDB	REGDET	Regional pipeline
OPSDB	SOCCPRO	Global pipeline
ARCHDB	AL1	Data archive
ARCHDB	AL2	Data archive

Table 5. Classified Database Accounts and Functions (Continued)

DATABASE NAME	ACCOUNT	FUNCTION SUPPORTED
ARCHDB	DETPRO	Data archive
ARCHDB	EVAL1	Data archive
ARCHDB	EVAL2	Data archive
ARCHDB	FAL	Data archive
ARCHDB	GLOBAL	Data archive
ARCHDB	HAL	Data archive
ARCHDB	HYDRODET	Data archive
ARCHDB	LFDET	Data archive
ARCHDB	LOOKUP	Data archive
ARCHDB	MONITOR	Data archive
ARCHDB	RAL1	Data archive
ARCHDB	RAL2	Data archive
ARCHDB	REGDET	Data archive
ARCHDB	SOCCPRO	Data archive

4.2.2.1 Multipurpose Support Accounts

4.2.2.1.1 GLOBAL Account

The GLOBAL account is available to all US NDC databases. It is the repository for the waveform file header parsed from incoming data streams by data acquisition applications like *DLParse*. In the unclassified OPSDB database, the GLOBAL account supports data acquisition and data forwarding to the archive database. In addition to these functions, the GLOBAL account in the classified OPSDB database supports data processing, data analysis, and data evaluation. In the archive databases, this account contains all of the archived data from the GLOBAL accounts on the data acquisition and data processing databases, as well as the **interval_files** table which supports data archiving. The GLOBAL account supports pipeline processing from the signal detection stage, when the incoming waveforms are reviewed by automated processes to detect the presence of a signal, through the last stage, when evaluators are generating event bulletins.

The GLOBAL account contains tables from the Continuous Data Subsystem (see Figure 14), the Message Subsystem (see Figure 15), the Data Archiving Subsystem (see Figure 16) and tables used for Distributed Processing Applications (see Figure 13). The **wfdisc** table is the largest and most often accessed table in the GLOBAL account. It contains the waveform file header data in the form of wfdisc records. The wfdisc record contains descriptive information about each

segment of waveform data in files stored on the filesystem. The *dir* column contains the path to the waveform file in the filesystem that is described by the wfdisc record.

The GLOBAL account also contains several other tables that are frequently accessed by data processing, data analysis and data evaluation applications. The **event**, **interval**, **lastid**, **remark** and **timestamp** tables are accessed by these applications directly through the GLOBAL account or through other accounts that have the appropriate data manipulation permissions on these tables. Read-only access to these tables is granted via the public role and public synonyms. Write access to these tables is granted by a role assigned to the pipeline processing accounts.

4.2.2.1.2 LOOKUP Account

The LOOKUP account contains all of the static information for all stations that send data to the US NDC System. The LOOKUP accounts in the OPSDB databases support data acquisition and processing. The LOOKUP accounts in the ARCHDB databases facilitate accessing archived data with interactive processing tools in a read-only fashion.

The LOOKUP account contains the tables described in Sections 4.1.3 and 4.1.4.2. The data in all the LOOKUP tables is made available to all US NDC applications and users on a read-only basis by way of the public role and public synonyms.

4.2.2.1.3 DEVNULL Account

The DEVNULL account contains tables that are place holders for applications that require tables not present in the account from which the application is running. This condition sometimes exists for interactive applications that are executed in the first interactive stage of a pipeline. Placing these placeholder tables in a separate account like DEVNULL ensures that no data can be inserted into these tables since permission to insert rows into the table is not granted to any other user.

4.2.2.1.4 MIGRATE Account

The MIGRATE account contains tables which hold the rules used by the *MigrateData* application. One set of rules tables is used to copy data from the classified OPSDB to the classified ARCHDB. A second set of rules tables is used for purging older data from OPSDB after it has been confirmed that the data has been successfully migrated to ARCHDB. The MIGRATE account exists only on the classified OPSDB.

Each rules table is an instantiation of the mig_rules schema element, which is used as a template for creation of rules tables by the mig_rules_cre.sql script. The rules tables for copying data to the ARCHDB are named date_<account name>_rules. The rules tables for purging older data from the OPSDB are named purge_<account name>_rules. There is one table of each type for the GLOBAL account, the MONITOR account, and each of the thirteen pipeline accounts.

4.2.2.1.5 MONITOR Account

The MONITOR account contains data regarding the scientific performance of data processing, and waveform data quality statistics generated by *PerfMon* and *DFX*. Figures 17 and 18 describe the tables in the MONITOR account and the relationships between the tables.

4.2.2.2 Pipeline Processing Accounts

The pipeline processing accounts provide the means by which data progresses through the data processing stages (pipelines) of the US NDC System. Additional detail about the types of processing which take place in the processing stages associated with each of these accounts can be found in *SAIC-02/3009, SSDD Phase 2 Build 1 (US NDC)*.

4.2.2.2.1 DETPRO Account

The DETPRO account contains data results of the Global pipeline automated station processing stage. The processes accessing the DETPRO account perform detection processing and form arrivals on seismic signals from individual stations as the first stage in the Global pipeline. The DETPRO account has tables from Figure 5, with the exception of the **hydro_** tables. The DETPRO account also contains pointers or synonyms to tables in the GLOBAL and LOOKUP accounts.

4.2.2.2.2 SOCCPRO Account

The SOCCPRO account contains the Global pipeline automated association processing stage results. In this stage of the Global pipeline, network processing is performed on the results from the previous pipeline stage. The tables illustrated in Figures 4 and 6 are contained in the SOCCPRO account, with the exception of the **amplitude** and **ampdescript** tables. These tables store station and network magnitude estimates. In addition to the synonyms for tables in GLOBAL and LOOKUP, the SOCCPRO account contains in_ synonyms, as described in Section 3.5.1.

4.2.2.2.3 AL1 Account

The AL1 account contains data results from the first pass of interactive analyst review. In this stage of the pipeline process, the analyst reviews events produced by the first two automated processing stages in the Global pipeline. The analyst may also identify new arrivals and form new events manually from late arriving data. This account contains tables from Figures 4, 5, 6, and 11. The AL1 account also contains pointers or synonyms to tables in the GLOBAL and LOOKUP accounts. In addition to the synonyms for tables in GLOBAL and LOOKUP, the AL1 account contains in_ and out_ synonyms, as described in Section 3.5.1.

4.2.2.2.4 AL2 Account

The AL2 account contains data results from the second interactive analyst review of events produced by the automated processing stages and AL1. This stage of pipeline processing further refines the events processed in AL1 and may identify new arrivals and form new events manually from late arriving data. This account contains the same tables as the AL1 account, synonyms for tables in GLOBAL and LOOKUP, and in_ and out_ synonyms as described in Section 3.5.1.

4.2.2.2.5 EVAL1 Account

The EVAL1 account contains data results from the first pass of evaluator review. In addition to containing results similar to those from the previous stages of the Global pipeline, the EVAL1 account also contains data from discrimination analysis. This data is stored in the tables shown in Figure 20. In addition to these tables, the EVAL1 account contains the same tables as AL2. EVAL1 also contains synonyms to tables in the GLOBAL and LOOKUP accounts and in_ and out_ synonyms, as described in Section 3.5.1.

4.2.2.2.6 EVAL2 Account

The EVAL2 account contains data results from the second evaluator review of the events produced by all previous stages of the Global pipeline. Processing is considered complete after this stage of processing is completed. EVAL2 contains all of the tables in EVAL1, synonyms to tables in the GLOBAL and LOOKUP accounts, and in_ and out_ synonyms as described in Section 3.5.1.

4.2.2.2.7 LFDET Account

The LFDET account contains data results from the first stage of automated processing in the Look-forward pipeline. It provides the same support to the Look-forward pipeline that the DETPRO and SOCCPRO accounts provide to the Global pipeline. LFDET contains the same tables as the DETPRO and SOCCPRO accounts and synonyms to the tables in the GLOBAL and LOOKUP accounts.

4.2.2.2.8 FAL Account

The FAL account contains data results from the first and only interactive analyst review of the events produced by the automated processing stage of the Look-forward pipeline. It contains the same tables as AL1 and AL2 with the exception of the **hydro_** tables. It also contains the synonyms to tables in the GLOBAL and LOOKUP accounts and in_ and out_ synonyms, as described in Section 3.5.1.

4.2.2.2.9 HYDRODET Account

The HYDRODET account is the first stage of automated processing in the hydroacoustic pipeline. It provides the same support to the Hydroacoustic pipeline that the DETPRO and SOCCPRO accounts provide to the Global pipeline. In addition to the tables in DETPRO and SOCCPRO, HYDRODET contains the **hydro_arrival** table and synonyms to the tables in the GLOBAL and LOOKUP accounts.

4.2.2.2.10 HAL Account

The HAL account contains data results from the first and only interactive analyst review of the events produced by the automated processing stage of the Hydroacoustic pipeline. It contains the same tables as AL1 and AL2, with the exception of the **hydro_arr_group** and **hydro_assoc** tables which will only be used in the Global pipeline. It also contains the synonyms to tables in the GLOBAL and LOOKUP accounts and in_ and out_ synonyms, as described in Section 3.5.1.

4.2.2.2.11 REGDET Account

The REGDET account is the first stage of automated processing in the Spotlight pipeline. It provides the same support to the Spotlight pipeline that the DETPRO and SOCCPRO accounts provide to the Global pipeline. REGDET contains the same tables as the DETPRO and SOCCPRO accounts and synonyms to the tables in the GLOBAL and LOOKUP accounts.

4.2.2.2.12 RAL1 Account

The RAL1 account contains data results from the first interactive analyst review of the events produced by the automated processing stage of the Spotlight pipeline. It contains the same tables as AL1 and AL2 with the exception of the **hydro_** tables. It also contains the synonyms to tables in the GLOBAL and LOOKUP accounts and in_ and out_ synonyms, as described in Section 3.5.1.

4.2.2.2.13 RAL2 Account

The RAL2 account contains data results from the second stage of interactive analyst review of the events produced by the automatic and first interactive analysis stages of the Spotlight pipeline. It contains the same set of tables and synonyms as the RAL1 account.

4.2.2.2.14 LOOKBACK Account

The LOOKBACK account is a special purpose account used by the evaluations and research personnel to interactively process data outside the normal pipeline processing timeline. If the data requested is less than 45 days old, it is copied into this account from the AL2 and DETPRO accounts in OPSDB; if the data requested is more than 45 days old, it is copied from ARCHDB. LOOKBACK is the lone exception to the use of in_ and out_ synonyms described in Section 3.5.1. LOOKBACK has a full range of out_ synonyms pointing to its own target tables.

However, LOOKBACK has actual tables corresponding to the in_ synonyms that would normally be present. These **in_** tables are clones of their equivalent target tables (i.e., **in_arrival** is a clone of **arrival**). These **in_** tables are used to hold the source data copied from the AL2 and DETPRO accounts before lookback processing is initiated.

LOOKBACK is also the only processing account which is not migrated to an equivalent LOOKBACK account in ARCHDB. Instead, when lookback processing is completed, the results are transferred to the EVAL2 tables. The results are then archived, along with other EVAL2 data.

4.2.2.3 Relational Database Management System (RDBMS) Accounts

The SYS, SYSTEM, OUTLN and DBSNMP accounts are present in all of the databases. These accounts are created by Oracle as part of the database creation process. They are used internally by the RDBMS to operate and maintain the database. The SYS account is the owner of the Oracle data dictionary objects. The SYS account is accessed by one of the UNIX shell scripts that runs as a CRON job (under the UNIX oracle account) to generate audit logs on database activity. SYSTEM is a standard account suitable for database administration. OUTLN owns the schema used to implement Optimizer Plan Stability. Database Simple Network Management Protocol (DBSNMP) is used to facilitate reporting database status to SNMP-based system management environments.

4.2.2.4 Maintenance Accounts

The Oracle account is used by the Oracle database administrators to maintain the user accounts and objects. It is also accessed by one of the UNIX shell scripts that runs as a CRON job (under the UNIX Oracle account) to generate daily tuning reports.

The Operator account was designed to be used as a read only interface to restricted Oracle data dictionary tables that normally can only be accessed by privileged users. This is accomplished by granting the select any table privilege to the operator account. Software maintainers can run scripts in the operator account that require read only access to some of the Oracle internal data dictionary objects.

4.2.2.5 Archive Database (ARCHDB) Accounts

The archive databases are used to store all data after it has been processed. For the unclassified archive database, data is stored until 180 days has elapsed since it was received on the US NDC System. On the classified archive database, data is stored forever.

With a few exceptions, the ARCHDB database accounts are replications of the corresponding accounts with the same names in the OPSDB databases. Since the archive is intended as a permanent historical record of results produced by the US NDC Systems, general read-write access is not permitted on the archive. Therefore, the read-write accounts present on OPSDB will not be created on ARCHDB. Only the MigrateData task and database administrators are given permission to alter the permanent contents of ARCHDB. The triggers enabled in some of

the OPSDB accounts are not present in the ARCHDB accounts. For performance purposes, there may be indexes on fields in the ARCHDB tables that may not exist in the OPSDB databases or vice versa.

4.2.3 Schema Element Definitions

The database schema is documented in the appendixes of this document:

- Appendix A contains table descriptions.
- Appendix B contains column descriptions.
- Appendix C contains view descriptions.
- Appendix D contains a list of accounts and tables.

4.3 Internal Design

Internal design includes a variety of aspects of Oracle installation and configuration design which are important to the operation of the databases, but which is not directly visible to the end user working with a schema. These aspects include:

- Products and options installed on the database
- Initialization parameters
- Configuration of rollback segments
- Configuration of redo logs
- Implementation of archive log mode

These aspects of the design cannot readily be determined until the database has been built on suitable hardware and appropriate configurations tested and confirmed. This information will be documented in an appendix to this document once it has been compiled.

4.4 Physical Design

This section summarizes the database servers and the databases that reside on each server. Detailed descriptions of the individual database tables and columns are described in the appendixes of this document.

There are four databases used by applications in the US NDC System. Two of these US NDC databases reside on the Unclassified System and the other two reside on the Classified System. In addition to the four US NDC databases, there are two instances of the OEM/RMAN repository database (identified as RCAT), one classified and one unclassified. The RCAT database holds the information needed to run the OEM and RMAN database administration tools and manage the US NDC databases. The RCAT database is used strictly to facilitate database administration and plays no role in the execution of US NDC applications. Table 6 lists the

databases described in this section. In the table, the databases are grouped together according to the system on which they reside. The table also provides a listing of the domains assigned to each of the databases and the names of the database servers on which the databases reside. The combination of the database name and the database domain constitute the fully qualified global service name identified in the local naming file (tnsnames.ora). A default domain of US NDC System is configured in the sqlnet.ora file so local users can reference the local databases as simply OPSDB or ARCHDB. Implementation of database domains and fully qualified service names will facilitate integration of the US NDC databases with other databases, as required, in AFTAC networks. The databases implemented at the Alt US NDC are identical to those implemented in the US NDC. The corresponding databases at each site have the same database name but are distinguished by the database domain as described previously.

Table 6. Summary of Database Instances

DATABASE NAME (SID)	FUNCTION	DATABASE DOMAIN	DATABASE SERVER [HARDWARE CONFIGURATION ITEM (HWCI)]
OPSDB	Unclassified Data Acquisition	USNDC ALTNDC	OPSDA (UDA) ALTDA
RCAT	OEM/RMAN Repository	USNDC ALTNDC	OPSDA (UDA) ALTDA
ARCHDB	Unclassified Data Archive	USNDC ALTNDC	OPSUARCH (UARC) ALTUARC
OPSDB	Classified Data Processing	USNDC ALTNDC	OPDBS (CA) ALTDBS
RCAT	OEM/RMAN Repository	USNDC ALTNDC	OPSDBS (CA) ALTDS
ARCHDB	Classified Data Archive	USNDC ALTNDC	OPSCARCH (CARC) ALTCARCH

4.4.1 Unclassified Data Acquisition Server

The OPSDA server hosts the classified instance of OPSDB that supports unclassified data acquisition. This includes cataloging all waveforms received from external sources and storing all of the associated alphanumeric data. Waveform descriptor (wfdisc) information is stored for 13 days. OPSDB also supports forwarding of data to the Classified System. The OPSDA server also hosts an instance of the OEM/RMAN repository database RCAT.

As previously discussed, machines capable of supporting multiple CPU domains are used to implement the US NDC System. The OPSDB database instance runs in one domain of a Sun Microsystems Enterprise 4800 server with 4 each 900 MHz Ultra Scaleable Processor Architecture (UltraSPARC) III CPUs and 8 gigabytes of memory. The same domain is used to execute the data acquisition software. Using the same domain to support both functions introduces little risk because unclassified OPSDB exists primarily to support data acquisition

and has very little extra load from ad hoc querying. The domain has two Sun A5200 disk arrays, configured as mirrors of one another, to accommodate both database filesystems and disk loop filesystems, which hold the 13 days of acquired data. Database filesystems are built in logical volumes distributed across the disks so as to equalize disk input/output (I/O) loading. Logical volumes are created to hold the following data structures:

- Datafiles assigned to RDBMS tablespaces
- Online transaction logs
- Datafiles assigned to the tablespaces holding US NDC data
- Datafiles assigned to the tablespaces holding indexes on US NDC data
- Daily backup files
- Archived transaction logs

Each schema owner account on OPSDB is assigned its own tablespace and associated datafile to hold its permanent schema elements. Each schema owner account also has a matching index tablespace with its own datafile.

4.4.2 Unclassified Archive Server

The OPSUARCH server hosts the unclassified instance of ARCHDB that supports the unclassified data archive. It stores all of the wfdisc records for all of the waveforms stored in the unclassified archive for 180 days. This archive is also used by AFTAC to respond to data requests from external customers.

The ARCHDB database instance runs in one domain of a Sun Microsystems Enterprise 4800 server with 2 each 900 megahertz (MHz) UltraSPARC III CPUs and 2 gigabytes of memory. The domain is dedicated solely to archiving functions. The domain has one Sun A5200 disk array to accommodate both database filesystems and archive disk cache filesystems, which hold the archived data that has not yet been committed to tape. Database filesystems are built in logical volumes distributed across the disks so as to equalize disk I/O loading. Logical volumes are created to hold the following data structures:

- Datafiles assigned to RDBMS tablespaces
- Online transaction logs
- Datafiles assigned to the tablespace holding US NDC data
- Datafiles assigned to the tablespace holding indexes on US NDC data
- Daily backup files
- Archived transaction logs

On ARCHDB, all indexes are created in one tablespace, which is assigned to one datafile. Likewise, all data tables are created in one tablespace, which is assigned to one datafile.

Multiple tablespaces dedicated to accounts were judged an unnecessary complication since only the MigrateData application and the waveform archive applications write data into ARCHDB.

4.4.3 Classified Processing Server

The OPSDBS server hosts the classified instance of OPSDB that supports classified data acquisition, analysis and evaluation. It stores the wfdisc information for 45 days, as well as all reference and alphanumeric data required for data processing and analysis. The account and table structure holds the results from automated and interactive data processing. It also supports the migration of data to the classified archive. The OPSDBS server also hosts an instance of the OEM/RMAN repository database RCAT.

The OPSDB database instance runs in one domain of a Sun Microsystems Enterprise 6800 server with 8 UltraSPARC III 900 MHz CPUs and 8 gigabytes of memory. Because this database supports a wide variety of activities, it requires a dedicated level of compute resources to ensure responsiveness. Accordingly, one of the four domains available on the 6800 is dedicated to OPSDB. The domain has two Sun A5200 disk arrays, configured as mirrors to one another, to accommodate both database filesystems and archive disk cache filesystems, which hold the archived data which has not yet been committed to tape. Database filesystems are built in logical volumes distributed across the disks so as to equalize disk I/O loading. Logical volumes are created to hold the following data structures:

- Datafiles assigned to RDBMS tablespaces
- Online transaction logs
- Datafiles assigned to the tablespace holding US NDC data
- Datafiles assigned to the tablespace holding indexes on US NDC data
- Daily backup files
- Archived transaction logs

Each schema owner account on OPSDB is assigned its own tablespace and associated datafile to hold its permanent schema elements. Each schema owner account also has a matching index tablespace with its own datafile. The read-write and read-only accounts for interactive analysis have a default tablespace called ANALYST where the non-permanent tables used for interprocess communication are created. Other US NDC accounts use the USERS tablespace as their default. Assignment of individual accounts to individual tablespaces was done to minimize the probability that internal data corruption in one account would cause a general failure of all processing on the US NDC System.

4.4.4 Classified Archive Server

The OPSCARCH server hosts the classified instance of ARCHDB that supports classified data archiving. All wfdisc records, interval data, and pipeline processing results are stored for the life of the US NDC System in ARCHDB accounts.

The ARCHDB database instance runs in one domain of a Sun Microsystems Enterprise 6800 server with 2 each 900 MHz UltraSPARC III CPUs and 2 gigabytes of memory. This domain is dedicated solely to archiving functions. The domain has one Sun A5200 disk array to accommodate both database filesystems and archive disk cache filesystems, which hold the archived data that has not yet been committed to tape. Database filesystems are built in logical volumes distributed across the disks so as to equalize disk I/O loading. Logical volumes are created to hold the following data structures:

- Datafiles assigned to RDBMS tablespaces
- Online transaction logs
- Datafiles assigned to the tablespace holding US NDC data
- Datafiles assigned to the tablespace holding indexes on US NDC data
- Daily backup files
- Archived transaction logs

On ARCHDB, all indexes are created in one tablespace, which is assigned to one datafile. Likewise, all data tables are created in one tablespace, which is assigned to one datafile. Multiple tablespaces dedicated to accounts were judged an unnecessary complication since only the MigrateData application and the archive applications write data into ARCHDB.

5. Detailed Design of Software Units Used for Database Access or Manipulation

See product documentation for the design details for specific software products that access the databases.

6. Requirements Traceability

See the US NDC traceability matrix in *SAIC 02/3009, System/Subsystem Design Description Phase 2 Build 1 (US NDC)*.

7. Notes

The following is a list of the acronyms and definitions used in this document.

ADSN	AFTAC Distributed Subsurface Network
AFTAC	Air Force Technical Applications Center
alphanumeric data	All American Standard Code for Information Interchange (ASCII) data, including parametric data, requests, and calibrations
Alt US NDC	Alternation United States National Data Center
API	Application programming interface
ARCHDB	Archive database
ARS	Analyst Review Station
ASCII	American Standard Code for Information Interchange
AutoDRM	Automatic Data Request Manager
BARM	Broad Area Regional Monitoring
CAS	Classified Acquisition Subsystem
CD	Continuous Data
COTS	Commercial Off-the-Shelf
CPU	Central processing unit
CSC	Computer software component
CSCI	Computer Software Configuration Item
CSS	Center for Seismic Study
DBDD	Database Design Description
DBI	Database Independent Interface
dbObj	Database object
DBSNMP	Database Simple Network Management Protocol
DII COE	Defense Information Infrastructure Common Operating Environment
DMS	Data Management System
DO	AFTAC Directorate of Operations
DoE	Department of Energy
GA	Global Association

GAFB	Goodfellow AFB
GDI	Generic Database Interface
GSE	Group of Scientific Experts
GSETT	Group of Scientific Experts Technical Test
GUI	Graphical User Interface
HQ	Headquarters
HWCI	Hardware Configuration Item
I/O	Input/output
IAW	In accordance with
IDC	International Data Centre
IMS	Intelligent Monitoring System
JDBC	Java Database Connectivity
LP	Long period
MHz	Megahertz
OCI	Oracle Call Interface
OEM	Oracle Enterprise Manager
OFA	Optimal Flexible Architecture
OLTP	On Line Transaction Processing
OPSDB	Operational Database
OS	Operating System
PAFB	Patrick AFB
PIDC	Prototype International Data Centre
QA	Quality Assurance
RDBMS	Relational Database Management System
RMAN	Recovery Manager
SNMP	Simple Network Management Protocol
SP	Short period
SPARC	Scaleable Processor Architecture
SQL	Structured query language
SRD	System Requirements Document
SSDD	System/Subsystem Design Description

TCP/IP	Transmission Control Protocol/Internet Protocol
US NDC	United States National Data Center
WAN	Wide Area Network
WGO	Working Group on Operations

Appendix A. Table Descriptions

This page is included in this document's electronic file as a placeholder for development of Table of Contents purposes only. The electronic version of this appendix is a separate file.

Appendix A. Table Descriptions

This appendix describes the ORACLE tables that comprise the US NDC database schema. The information given here, together with that provided in Appendix B, constitutes the data dictionary. There is an entry for each table. Within the entry, the name of the database table appears first, followed by a description of the purpose and use of the tables. Below the description is a listing of the columns, in the order in which they are defined in the tables. The storage type column gives the actual ORACLE datatype for the column in question. The external format and character positions column are provided for the convenience of database users who wish to transfer data between the ORACLE database tables and flat files.

A1. Table Categories

The field immediately following the table definition is the category field. The categories represent generalized system functions that the database tables support and include the following:

- Fundamental
- Fundamental Reference
- Continuous Data Subsystem
- Distributed Processing
- Network Processing
- Event Discrimination
- Event Processing
- Hydroacoustic Processing
- Hydroazimuth Processing
- Map
- Performance Monitoring
- System Monitoring
- Data Administration
- Data Archiving
- Data Migration
- Message Subsystem

A2. Computer Software Configuration Items (CSCIs)

The field following the table category is the CSCI field. One or more of the following CSCIs are listed for each table:

- Data Services
- Data Management
- Distributed Application Control System
- Automatic Processing
- Interactive Processing
- Performance Monitoring
- Tuning Tools

The CSCI fields in the table description indicate that an application in that CSCI accesses the table through one of the interfaces defined in Section 3.6.

A3. Column Categories

Below the category definition of the database table itself are categories for the columns in the database table. The columns of the database table are categorized as keys and data. Key columns link database tables. Data columns, the reason that database tables exist, are split into three categories: Descriptive, Measurement, and Administrative. The following explains the format used in the entries:

Keys:	Primary	The columns which, when taken together, uniquely identify a row in the database table.
	Alternate	Other columns that also uniquely identify a row and may be used as surrogates for primary keys.
	Foreign	Primary keys in another database table.
Data:	Descriptive	Qualitative columns
	Measurement	Quantitative columns
	Administration	Columns used for database administration

Keys provide the links by which database tables are joined. The following definitions explain the types of keys:

- Primary key: Uniquely identifies a row in a database table (often the concatenation of several columns). For example, every **origin** record is unique by *lat*, *lon*, *depth*, and *time*
- Alternate key: Also uniquely identifies a row in a database table and may be used as a surrogate for the primary key. For example, *orid* may also be used as the primary key in the **origin** database table. Alternate keys in the US NDC databases are always id-based.
- Foreign key: Refers to another database table's primary key. For example, *evid* is a foreign key in the **origin** database table but is the primary key in the **event** database table; *commid* is a foreign key in many of the database tables but is the primary key in the **remark** database table.

A4. Conventions

This section uses geographical and typographical conventions as described in Table A1.

Table A1. Typographical Conventions

ELEMENT	APPEARANCE	EXAMPLE
Database table Database table and columns, when written in the dot notation	Bold	**dataready** **prodtrack.status**
Database columns Processes, software units, and libraries Titles of documents	Italics	*Status* *ARS, libpar* *GA Subsystem Software*
Value of a key or component of a key	Courier font	`orid`

A5. Table Definitions

Tables A2 through A76 provide table definitions. Tables in parentheses are clones of schema tables used by some applications as storage areas for intermediate results. See Section 3.9 for more information on the use of clones in the US NDC databases.

affiliation (stanet)

The **affiliation** table groups stations into networks. The **stanet** table is used for Distributed Processing. It contains station to array mapping.

Table A2. affiliation (stanet)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*net*	varchar2(8)	a8	1-8	Unique network identifier
2	*sta*	varchar2(6)	a6	10-15	Station identifier
3	*time*	float(53)	f17.5	17-33	Starting *time* for station in network
4	*endtime*	float(53)	f17.5	35-51	*endtime* for station in network
5	*lddate*	date	a19	53-71	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*net/sta/time*
Data:	Descriptive	*net, sta*
	Measurement	*time, endtime*
	Administrative	*lddate*

alphasite

The **alphasite** table is used for tracking continuous data connections by the *DLMan* application. For a given station, there must be a row in the table for each address from which the station may send continuous data. The columns p*refdlid* and *prefport* describe the preferred *DLMan* connection for the station. *DLMan* fills in *dlid* and *time* while a station is actively connected. The column *dlid* indicates to which *dlid* the station is currently connected (may be different from *prefdlid*) and the column *time* is the system time for the last activity on the station's connection (not the time of any data received). For stations that are not connected, *time* and *dlid* are zero.

Table A3. alphasite

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*address*	varchar2(16)	a16	8-23	Source internet address
3	*prefdlid*	number(8)	i8	25-32	*dlid* for preferred *DLMan*
4	*prefport*	number(8)	i8	34-41	Preferred network port
5	*dlid*	number(8)	i8	43-50	*dlid* handling station
6	*time*	float(53)	f17.5	52-68	Clock time of most recent activity
7	*commid*	number(9)	i9	70-78	Comment identifier
8	*lddate*	date	a19	80-98	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*sta/address*
	Foreign	*dlid, commid*
Data:	Descriptive	*sta, address, prefdlid, prefport, dlid*
	Measurement	*time*
	Administrative	*lddate*

amp3c

The **amp3c** table contains amplitude measurements made on three-component data for a specific detection.

Table A4. amp3c

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*arid*	number(9)	i9	1-9	Arrival identifier
2	*cfreq*	float(24)	f7.2	11-17	Center frequency of filter band amplitude is measured on
3	*vamp*	float(24)	f11.2	19-29	Vertical amplitude
4	*vsnr*	float(24)	f10.2	31-40	Vertical signal-to-noise ratio
5	*hamp*	float(24)	f11.2	42-52	Horizontal amplitude
6	*hsnr*	float(24)	f10.2	54-63	Horizontal signal-to-noise ratio
7	*htov*	float(24)	f10.2	65-74	Horizontal-to-vertical amplitude ratio
8	*rid*	varchar2(8)	a8	76-83	Recipe identifier
9	*lddate*	date	a19	85-103	Load date

Category:	Fundamental	
CSCI(s)	Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*arid/cfreq*
Data:	Measurement	*rid, vamp, vsnr, hamp, hsnr, htov*
	Administrative	*lddate*

ampdescript

The **ampdescript** table contains descriptions of how amplitude measurements in **amplitude** were made.

Table A5. ampdescript

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*amptype*	varchar2(8)	a8	1-8	Amplitude measure descriptor
2	*toff*	float(24)	f6.2	10-15	Offset from theoretical or observed arrival time
3	*tlen*	float(24)	f10.3	17-26	Duration of measurement window
4	*gvlo*	float(24)	f5.2	28-32	Low group velocity for measurement window (km/sec)
5	*gvhi*	float(24)	f5.2	34-38	High group velocity for measurement window kilometers per second (km/sec)
6	*mtype*	varchar2(8)	a8	40-47	Measurement type
7	*ampdescr*	varchar2(255)	a255	49-303	Description
8	*lddate*	date	a19	305-323	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*amptype*
Data:	Descriptive	*amptype, mtype,ampdescr*
	Measurement	*toff, tlen, gvlo, gvhi*
	Administrative	*lddate*

amplitude

The **amplitude** table contains arrival-based and origin-based amplitude measurements. The amplitude measurement is described in **ampdescript**.

Table A6. amplitude

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*ampid*	number(9)	i9	1-9	Amplitude identifier
2	*arid*	number(9)	i9	11-19	Arrival identifier
3	*parid*	number(9)	i9	21-29	Predicted arrival identifier
3	*chan*	varchar2(8)	a8	31-38	Channel code
4	*amp*	float(24)	f11.2	40-50	Amplitude
5	*per*	float(24)	f7.2	52-58	Period(s)
6	*snr*	float(24)	f10.2	60-69	Signal-to-noise ratio
7	*amptime*	float(53)	f17.5	71-87	Time of amplitude measure
8	*time*	float(53)	f17.5	89-105	Start time of measurement window
9	*duration*	float(24)	f7.2	107-113	Duration of measurement window
10	*deltaf*	float(24)	f7.3	115-121	Sample interval width
11	*amptype*	varchar2(8)	a8	123-130	Amplitude measure descriptor
12	*units*	varchar2(15)	a15	132-146	Units
13	*clip*	varchar2(1)	a1	148-148	Clipped flag
14	*inarrival*	varchar2(1)	a1	150-150	y or n flag indicating if amplitude (*amp*) is the same as the *amp* in the arrival table

Table A6. amplitude (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
15	*auth*	varchar2(15)	a15	152-166	Author
16	*lddate*	date	a19	168-186	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*ampid*
	Foreign	*arid, amptype*
Data:	Descriptive	*chan, amptype, units, inarrival, parid*
	Measurement	*amp, per, snr, amptime, time, duration, deltaf, clip*
	Administrative	*auth, lddate*

aoi

The **aoi** table contains geographic characteristics of a particluar region of the Earth, based on four criteria: *aoi_geochar*, *depth_geochar*, *seismic_geochar*, and *terrain_geochar*.

Table A7. aoi

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*lat*	float(53)	f11.6	1-11	Latitude
2	*lon*	float(53)	f11.6	13-23	Longitude
3	*aoi_geochar*	varchar2(1)	a1	25-25	Area of interest geographic region characteristic
4	*depth_geochar*	varchar2(1)	a1	27-27	Depth geographic region characteristic
5	*seismic_geochar*	varchar2(1)	a1	29-29	Seismic geographic region characteristic
6	*terrain_geochar*	varchar2(1)	a1	31-31	Terrain geographic region characteristic
7	*ondate*	number(8)	i8	33-40	Julian on date
8	*offdate*	number(8)	i8	42-49	Julian off date
9	*lddate*	date	a19	51-69	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Management	
Keys:	Primary	*lat, lon*
Data:	Descriptive	*lat, lon, aoi_geochar, depth_geochar, seismic_geochar, terrain_geochar*
	Measurement	*ondate, offdate*
	Administrative	*lddate*

apma

The **apma** table contains results of particle motion analysis for a specific detection.

Table A8. apma

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*phase*	varchar2(8)	a8	1-8	Phase
2	*arid*	number(9)	i9	10-18	Arrival identifier
3	*freq*	float(24)	f7.2	20-26	Frequency
4	*snr*	float(24)	f10.2	28-37	Signal-to-noise ratio
5	*ampp*	float(24)	f7.2	39-45	P-phase amplitude
6	*amps*	float(24)	f7.2	47-53	S-phase amplitude
7	*amplr*	float(24)	f7.2	55-61	Rayleigh-phase amplitude
8	*rect*	float(24)	f7.3	63-69	Rectilinearity
9	*plans*	float(24)	f7.2	71-77	S-phase planarity
10	*planlr*	float(24)	f7.2	79-85	Rayleigh-phase planarity
11	*hvratp*	float(24)	f7.2	87-93	P-phase horizontal-to-vertical ratio
12	*hvrat*	float(24)	f7.2	95-101	S-phase horizontal-to-vertical ratio
13	*hmxmn*	float(24)	f7.2	103-109	Maximum-to-minimum horizontal ratio
14	*inang3*	float(24)	f7.2	111-117	Short-axis incidence angle
15	*seazp*	float(24)	f7.2	119-125	P-phase observed azimuth
16	*seazs*	float(24)	f7.2	127-133	S-phase observed azimuth
17	*seazlr*	float(24)	f7.2	135-141	Rayleigh-phase observed azimuth
18	*inang1*	float(24)	f7.2	143-149	Long-axis incidence angle
19	*pphasetime*	float(53)	f17.5	151-167	P-phase extraction time

Table A8. apma (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
20	*sphasetime*	float(53)	f17.5	169-185	S-phase extraction time
21	*auth*	varchar2(15)	a15	187-201	Author
22	*apmarid*	number(8)	i8	203-210	apma recipe identifier
23	*commid*	number(9)	i9	212-220	Comment identifier
24	*lddate*	date	a19	222-240	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*arid*
	Foreign	*commid*
Data:	Descriptive	*phase*
	Measurement	*freq, snr, ampp, amps, amplr, rect, plans, planlr, hvratp, hvrat, hmxmn, inang3, seazp, seazs, seazlr, inang1, pphasetime, sphasetime, amparid*
	Administrative	*auth, lddate*

arrival

The **arrival** table contains summary information about arrivals.

Table A9. arrival

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*time*	float(53)	f17.5	8-24	Epoch time
3	*arid*	number(9)	i9	26-34	Arrival identifier
4	*jdate*	number(8)	i8	36-43	Julian date
5	*stassid*	number(8)	i8	45-52	Arrival group indentification
6	*chanid*	number(8)	i8	54-61	Instrument identifier
7	*chan*	varchar2(8)	a8	63-70	Channel code
8	*iphase*	varchar2(8)	a8	72-79	Reported phase
9	*stype*	varchar2(1)	a1	81-81	Signal type
10	*deltim*	float(24)	f6.3	83-88	Time uncertainty
11	*azimuth*	float(24)	f7.2	90-96	Observed azimuth
12	*delaz*	float(24)	f7.2	98-104	Azimuth uncertainty
13	*slow*	float(24)	f7.2	106-112	Observed slowness, seconds/degree
14	*delslo*	float(24)	f7.2	114-120	Slowness uncertainty
15	*ema*	float(24)	f7.2	122-128	Emergence angle
16	*rect*	float(24)	f7.3	130-136	Rectilinearity
17	*amp*	float(24)	f11.2	138-148	Amplitude, instrument corrected
18	*per*	float(24)	f7.2	150-156	Period
19	*logat*	float(24)	f7.2	158-164	Log (amp/per)
20	*clip*	varchar2(1)	a1	166-166	Clipped flag
21	*fm*	varchar2(2)	a2	168-169	First motion
22	*snr*	float(24)	f10.2	171-180	Signal-to-noise ratio
23	*qual*	varchar2(1)	a1	182-182	Signal onset quality

Table A9. arrival (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
24	*auth*	varchar2(15)	a15	184-198	Author
25	*commid*	number(9)	i9	200-208	Comment identifier
26	*lddate*	date	a19	210-228	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/time*
	Alternate	*arid*
	Foreign	*chanid, commid*
Data:	Descriptive	*sta, chan, iphase, stype*
	Measurement	*time, jdate, deltim, azimuth, delaz, slow, delslo, ema, rect, amp, per, logat, clip, fm, snr, qual*
	Administrative	*auth, lddate*

assoc (assoc_ga)

The **assoc** table contains information that connects arrivals (entries in the **arrival** table) to a particular origin. The **assoc_ga** table is used by the Global Association (*GA*) application to store temporary associations.

Table A10. assoc (assoc_ga)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*arid*	number(9)	i9	1-9	Arrival identifier
2	*orid*	number(9)	i9	11-19	Origin identifier
3	*sta*	varchar2(6)	a6	21-26	Station code
4	*phase*	varchar2(8)	a8	28-35	Associated phase
5	*belief*	float(24)	f4.2	37-40	Phase confidence
6	*delta*	float(24)	f8.3	42-49	Station-to-event distance
7	*seaz*	float(24)	f7.2	51-57	Station-to-event azimuth
8	*esaz*	float(24)	f7.2	59-65	Event-to-station azimuth
9	*timeres*	float(24)	f8.3	67-74	Time residual
10	*timedef*	varchar2(1)	a1	76-76	Time = defining (`d`), nondefining (`n`)
11	*azres*	float(24)	f7.1	78-84	Azimuth residual
12	*azdef*	varchar2(1)	a1	86-86	Azimuth = defining (`d`), nondefining (`n`)
13	*slores*	float(24)	f7.2	88-94	Slowness residual
14	*slodef*	varchar2(1)	a1	96-96	Slowness = defining (`d`), nondefining (`n`)
15	*emares*	float(24)	f7.1	98-104	Incidence angle residual
16	*wgt*	float(24)	f6.3	106-111	Location weight

Table A10. assoc (assoc_ga) (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
17	*vmodel*	varchar2(15)	a15	113-127	Velocity model
18	*commid*	number(9)	i9	129-137	Comment identifier
19	*lddate*	date	a19	139-157	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*arid/orid*
	Foreign	*commid*
Data:	Descriptive	*sta, phase, belief, wgt, vmodel*
	Measurement	*delta, seaz, esaz, timeres, timedef, azres, azdef, slores, slodef, emares*
	Administrative	*lddate*

bull_comp

The **bull_comp** table contains results from the *BullComp* application of the comparison of two seismic bulletins. The information summarizes the differences between event solutions that share common associated arrivals or (if no arrival information is available) overlapping locations and time uncertainties.

Table A11. bull_comp

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid1*	number(9)	i9	1-9	Origin identifier from bulletin 1
2	*orid2*	number(9)	i9	11-19	Origin identifier from bulletin 2
3	*ddist*	float(24)	f8.3	21-28	Difference in distance
4	*ddepth*	float(24)	f6.1	30-35	Difference in depth
5	*dtime*	float(24)	f8.3	37-44	Difference in epoch time
6	*ndef1*	number(8)	i8	46-53	Number of time-defining phases for *orid1*
7	*ndef2*	number(8)	i8	55-62	Number of time-defining phases for *orid2*
8	*dndef*	number(8)	i8	64-71	Difference in number of time-defining phases
9	*narr1*	number(8)	i8	73-80	Number of associated arrivals for *orid1*
10	*narr2*	number(8)	i8	82-89	Number of associated arrivals for *orid2*
11	*dnarr*	number(8)	i8	91-98	Difference in number of associated arrivals
12	*nmatch*	number(8)	i8	100-107	Number of matching arrivals (defining/ nondefining)
13	*ndef1arr2*	number(8)	i8	109-116	Number of defining arrivals for *orid1* that are arrivals (either defining or nondefining) for *orid2*
14	*ndef2arr1*	number(8)	i8	118-125	Number of defining arrivals for *orid2* that are arrivals (defining/ nondefining) for *orid1*

Table A11. bull_comp (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
15	*asstr*	varchar2(1)	a1	127-127	Association strength (`s` or `w`)
16	*bulletins*	varchar2(64)	a64	129-192	Which bulletins are being compared
17	*time*	float(53)	f17.5	194-210	Epoch start time
18	*endtime*	float(53)	f17.5	212-228	End time of bulletin comparison
19	*lddate*	date	a19	230-248	Load date

Category:	System Monitoring	
CSCI(s)	Data Management, Performance Monitoring	
Keys:	Primary	*orid1/orid2*
Data:	Descriptive	*asstr, bulletins*
	Measurement	*ddist, ddepth, dtime, ndef1, ndef2, dndef, narr1, narr2, dnarr, nmatch, ndef1arr2, ndef2arr1, time, endtime*
	Administrative	*lddate*

calibrate

The **calibrate** table contains default calibration values for stations that have periodic calibration and where the data is not adjusted to a nominal *calib* value. For stations sending calibration values in the data stream, the **calibrate** table contains a historic record of the values as they change over time.

Table A12. calibrate

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*calib*	float(24)	f16.6	17-32	Calibration
4	*calper*	float(24)	f16.6	34-49	Calibration period
5	*time*	float(53)	f17.5	51-67	Epoch start time
6	*endtime*	float(53)	f17.5	69-85	Epoch end time
7	*chanid*	number(8)	i8	87-94	Channel identifier
8	*lddate*	date	a19	96-114	Load date

Category:	Continuous Data Services	
CSCI(s)	Data Services, Interactive Processing, Performance Monitoring	
Keys:	Primary	*sta/chan/time/endtime*
	Foreign	*chanid*
Data:	Measurement	*calib, calper*
	Administrative	*lddate*

chan_groups

The **chan_groups** table is used to indicate which *sta/chan* pairs belong to a given *class/name* (**wfactivity**) group.

Table A13. chan_groups

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*class*	varchar2(16)	a16	1-16	Type of interval
2	*name*	varchar2(20)	a20	18-37	Name of interval
3	*sta*	varchar2(6)	a6	39-44	Station
4	*chan*	varchar2(8)	a8	46-53	Channel code
5	*duration*	float(24)	f7.2	55-61	Duration in seconds of the time region
6	*inwfactivity*	number(1)	I1	63-63	Is this *class/name/duration* in **wfactivity**
7	*ondate*	number(8)	i8	65-72	On date
8	*offdate*	number(8)	i8	74-81	Off date
9	*lddate*	date	a19	83-101	Load date

Category:	Data Archiving	
CSCI(s)	Data Management	
Keys:	Primary	*class/name/sta/chan*
Data:	Descriptive	*sta, chan, duration, inwfactivity, ondate, offdate*
	Administrative	*lddate*

channame

The **channame** table provides mapping between channel and station names.

Table A14. channame

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*extern_sta*	varchar2(6)	a6	1-6	External station name
2	*extern_chan*	varchar2(8)	a8	8-15	External channel name
3	*extern_auth*	varchar2(20)	a20	17-36	External authority using this name
4	*intern_sta*	varchar2(6)	a6	38-43	Internal station name
5	*intern_chan*	varchar2(8)	a8	45-52	Internal channel name
6	*intern_chanid*	number(8)	i8	54-61	Internal channel ID
7	*commid*	number(9)	i9	63-71	Comment identifier
8	*lddate*	date	a19	73-91	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management, Interactive Processing	
Keys:	Primary	*extern_sta/extern_chan*
	Foreign	*intern_sta/intern_chan, intern_chanid, commid*
Data:	Descriptive	*extern_sta, extern_chan, intern_sta, intern_chan, intern_chanid*
	Administrative	*lddate*

colordisc

The **colordisc** table links a unique *colormapid* to a colormap name and disk file. It is used by the *Map* application.

Table A15. colordisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*colormapid*	number(8)	i8	1-8	Colormap identifier
2	*dfile*	varchar2(32)	a32	10-41	Data filename
3	*dir*	varchar2(64)	a64	43-106	Directory
4	*colormapname*	varchar2(64)	a64	108-171	Colormap name
5	*lddate*	date	a19	173-191	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*colormapid*
Data:	Descriptive	*colormapname, dfile, dir*
	Administrative	*lddate*

datacollected

The **datacollected** table records information for *PerfMon* to determine if image generation can be performed

Table A16. datacollected

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*topic*	varchar2(8)	a8	1-8	Type of data processing
2	*database1*	varchar2(15)	a15	10-24	Name of first source database
3	*database2*	varchar2(15)	a15	26-40	Name of second source database
4	*processing*	varchar2(16)	a16	42-57	*PerfMon* mode
5	*bullcomp*	varchar2(30)	a30	59-88	*Bullcomp* description
6	*region*	varchar2(30)	a30	90-119	Limits of geographic region
7	*time*	float(53)	f17.5	121-137	Start time of bulletin comparison
8	*endtime*	float(53)	f17.5	139-155	End time of bulletin comparison
9	*lddate*	date	a19	157-175	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	NA
Data:	Descriptive	*topic, database1, database2, processing, bullcomp, region*
	Measurement	*time, endtime*
	Administrative	*lddate*

datadays

The **datadays** table contains the days and times for which analysis has been completed.

Table A17. datadays

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*pmdescr*	varchar2(64)	a64	1-64	Description of *PerfMon* state
2	*jdate*	number(8)	i8	66-73	Julian date
3	*time*	float(53)	f17.5	75-91	Epoch time of start of dataday
4	*endtime*	float(53)	f17.5	93-109	Epoch time of end of dataday
5	*lddate*	date	a19	111-129	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*jdate*
Data:	Measurement	*jdate, time, endtime*
	Administrative	*lddate*
	Descriptive	*pmdescr*

datauser

The **datauser** table tracks authorized users of the Message and Subscription Subsystems. Each user is identified by a (unique) *username* and *domain*, which must match all electronic mail (e-mail) headers. The *priority* column specifies the class of user, and *servicetime* is the last time a request from the user was processed. *Priority* and *servicetime* are considered when selecting the order in which requests will be processed. The status can either be active or inactive.

Table A18. datauser

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*userid*	number(8)	i8	1-8	Identifier for the user
2	*pocid*	number(8)	i8	10-17	Point of contact identifier
3	*username*	varchar2(24)	a24	19-42	User name from the incoming subscription message
4	*domain*	varchar2(48)	a48	44-91	Domain name from the incoming subscription message
5	*msgtype*	varchar2(16)	a16	93-108	Message type
6	*userstatus*	varchar2(24)	a24	110-133	Status of this user
7	*priority*	number(2)	I2	135-136	User's priority
8	*commid*	number(9)	i9	138-146	Comment identifier
9	*emaillimit*	number(8)	i8	148-155	Maximum size of message (in bytes) that will be delivered via e-mail
10	*servicetime*	float(53)	f17.5	157-173	Last time a request from that user was serviced
11	*lddate*	date	a19	175-193	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*userid*
	Foreign	*commid*
Data:	Descriptive	*username, domain, emaillimit, pocid*
	Measurement	*userstatus, priority, servicetime*
	Administrative	*lddate*

detection

The **detection** table contains summary information about waveform.

Table A19. detection

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*arid*	number(9)	i9	1-9	Arrival identifier
2	*jdate*	number(8)	i8	11-18	Julian date
3	*time*	float(53)	f17.5	20-36	Epoch time
4	*sta*	varchar2(6)	a6	38-43	Station code
5	*chan*	varchar2(8)	a8	45-52	Channel code
6	*bmtyp*	varchar2(4)	a4	54-57	Beam type
7	*sproid*	number(8)	i8	59-66	Signal processor identifier
8	*cfreq*	float(24)	f7.2	68-74	Center frequency
9	*seaz*	float(24)	f7.2	76-82	Observed azimuth
10	*delaz*	float(24)	f7.2	84-90	Azimuth uncertainty
11	*slow*	float(24)	f7.2	92-98	Observed slowness, seconds per kilometer (sec/km)
12	*delslo*	float(24)	f7.2	100-106	Slowness uncertainty
13	*snr*	float(24)	f10.2	108-117	Signal-to-noise ratio
14	*stav*	float(24)	f11.5	119-129	Short-term average
15	*fstat*	float(24)	f5.2	131-135	f-statistic
16	*deltim*	float(24)	f6.3	137-142	Time uncertainty

Table A19. detection (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
17	*bandw*	float(24)	f7.3	144-150	Bandwidth
18	*fkqual*	number(4)	i4	152-155	f-k quality
19	*commid*	number(9)	i9	157-165	Comment identifier
20	*lddate*	date	a19	167-185	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/time*
	Alternate	*arid*
	Foreign	*commid*
Data:	Descriptive	*sta, chan, bmtyp, sproid*
	Measurement	*jdate, time, cfreq, seaz, delaz, slow, delslo, snr, stav,fstat, deltim, bandw, fkqual*
	Administrative	*lddate*

discrimuse

The **discrimuse** table contains the use/nonuse of station data in discriminant voting. It identifies, for each station associated to the origin, the use or non-use of that station's data in the discriminant vote for six different discriminants. The votes are then combined to determine the overall event classification (see the **discrimvote** table).

Table A20. discrimuse

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identifier
2	*discrimtype*	varchar2(10)	a10	11-20	Discriminant type
3	*sta*	varchar2(6)	a6	22-27	Station
4	*discrim_flag*	varchar2(1)	a1	29-29	Discriminant is used in event classification for the origin and station (T, F)
5	*commid*	number(9)	i9	31-39	Comment identifier
6	*lddate*	date	a19	41-59	Load date

Category:	Event Discrimination	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*orid/discrimtype/sta*
	Foreign	*commid*
Data:	Descriptive	*discrimtype, sta, discrim_flag*
	Administrative	*lddate*

discrimvote

The **discrimvote** table identifies the vote value for each of six discriminants (*discrimtypes*). These votes are combined to determine the overall event classification. When the vote is overridden, it includes evaluator's comments.

Table A21. discrimvote

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identifier
2	*discrimtype*	varchar2(10)	a10	11-20	Discriminant type
3	*vote*	varchar2(1)	a1	22-22	Overall vote for the discriminant factor
4	*commid*	number(9)	i9	24-32	Comment identifier
5	*override*	number(8)	i8	34-40	Evaluator override vote
6	*eval_comment*	varchar2(22)	a22	42-64	Evaluator comments on override
7	*lddate*	date	a19	66-84	Load date

Category:	Event Discrimination	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*orid/discrimtype*
	Foreign	*commid*
Data:	Descriptive	*discrimtype, vote, override, eval_comment*
	Administrative	*lddate*

dlfile

The **dlfile** table describes the files used in the diskloops managed by the Continuous Data Services Subsystem.

Table A22. dlfile

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*dir*	varchar2(64)	a64	1-64	Directory name
2	*dfile*	varchar2(32)	a32	66-97	Filename
3	*machine*	varchar2(32)	a32	99-130	Machine name
4	*partition*	varchar2(64)	a64	132-195	Disk partition name
5	*dfid*	number(9)	i9	197-205	Diskloop file identifier
6	*inloop*	varchar2(1)	a1	207-207	File is part of a diskloop = y/n
7	*full*	varchar2(1)	a1	209-209	File is full = y/n
8	*archived*	varchar2(1)	a1	211-211	File is archived = y/n/a
9	*length*	number(10)	i10	213-222	Length of file, bytes
10	*tlen*	float(24)	f10.3	224-233	Length of file, seconds
11	*time*	float(53)	f17.5	235-251	Start time
12	*reaptime*	float(53)	f17.5	253-266	Clock time for expiration
13	*sta*	varchar2(6)	a6	271-276	Station name
14	*chan*	varchar2(8)	a8	278-285	Channel code
15	*chanid*	number(8)	i8	287-294	Channel identifier
16	*dlid*	number(8)	i8	296-303	Diskloop manager identifier
17	*commid*	number(9)	i9	305-313	Comment identifier
18	*lddate*	date	a19	315-333	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*dir/dfile*
	Alternate	*dfid*
	Foreign	*chanid, dlid, commid*
Data:	Descriptive	*dir, dfile, machine, partition, inloop, full, archived, length, tlen, time, sta, chan*
	Administrative	*reaptime, lddate*

dlman

The **dlman** table keeps track of currently running *DLMan* instances. The column m*achine* is the host on which this *dlid* runs (it may not run elsewhere). The column r*unning* indicates whether that *DLMan* is currently operational. The table also provides the ports that this *dlid* is currently using to listen to other processes.

Table A23. dlman

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*dlid*	number(8)	i8	1-8	Diskloop manager identifier
2	*machine*	varchar2(32)	a32	10-41	Machine name
3	*running*	varchar2(1)	a1	43-43	*Dlman* running = y/n
4	*connmanport*	number(6)	i6	45-50	Connman port
5	*controlport*	number(6)	i6	52-57	Datacontrol port
6	*archiveport*	number(6)	i6	59-64	Archiver port
7	*forwardport*	number(6)	i6	66-71	Forwarder port
8	*commid*	number(9)	i9	73-81	Comment identifier
9	*lddate*	date	a19	82-101	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*dlid*
	Foreign	*commid*
Data:	Descriptive	*machine, running, connmanport, controlport, archiveport, forwardport*
	Administrative	*lddate*

ev_summary (an_summary, ex_summary)

The **ev_summary** (**an_summary, ex_summary**) table contain statistical summary analysis of expert system solutions from the *ExAnComp* application.

Table A24. ev_summary (an_summary, ex_summary)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identifier of analyst event
2	*nearsta*	varchar2(6)	a6	11-16	Code for nearest station
3	*neardist*	float(24)	f8.3	18-25	Distance to closest station
4	*nearaz*	float(24)	f7.2	27-33	Azimuth from nearest station
5	*refid*	number(8)	i9	35-43	Identifier of nearest reference point
6	*refdist*	float(24)	f8.3	45-52	Distance to nearest reference point
7	*refaz*	float(24)	f7.2	54-60	Azimuth to nearest reference point
8	*grn*	number(8)	i8	62-69	Geographic region number
9	*nsta*	number(8)	i8	71-78	Number of recording stations
10	*lsta*	number(8)	i8	80-87	Number of local observations
11	*asta*	number(8)	i8	89-96	Number of regional array observations
12	*rsta*	number(8)	i8	98-105	Number of non-array regional observations
13	*tsta*	number(8)	i8	107-114	Number of teleseismic observations
14	*ndef*	number(4)	I4	116-119	Number of time-defining phases
15	*adef*	number(8)	i8	121-128	Number of associated nondefining phases
16	*primp*	number(8)	i8	130-137	Number of primary time-defining phases used for location

Table A26. ev_summary (an_summary, ex_summary) (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
17	*secondp*	number(8)	i8	139-146	Number of secondary phases used for location
18	*depthp*	number(8)	i8	148-155	Number of depth phases
19	*lddate*	date	a19	157-175	Load date

Category:	System Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*orid*
	Foreign	*grn, refid*
Data:	Descriptive	*nearsta, grn*
	Measurement	*neardist, nearaz, refdist, refaz, nsta, lsta, asta, rsta, tsta, ndef, adef, primp, secondp, depthp*
	Administrative	*lddate*

event

The **event** table contains a list of events. Multiple origins may be defined for any one event. The column *prefor* points to the preferred origin.

Table A25. event

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*evid*	number(8)	i8	1-8	Event identifier
2	*evname*	varchar2(32)	a32	10-41	Event name
3	*prefor*	number(9)	i9	43-50	Preferred origin
4	*auth*	varchar2(15)	a15	52-66	Source/originator
5	*commid*	number(8)	i8	68-75	Comment identifier
6	*lddate*	date	a19	77-95	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*evid*
	Foreign	*prefor, commid*
Data:	Descriptive	*evname, prefor*
	Administrative	*auth, lddate*

event_control

The **event_control** table contains event location and magnitude control parameters. This information acts as an archive of the specific user-defined controls that were used to determine the location and magnitude of a given *orid*. The table also includes two measurement columns (*cov_sm_axes* and *cov_depth_time*) that allow the coverage ellipse to be determined from the confidence ellipse axes.

Table A26. event_control

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identification
2	*evid*	number(9)	i9	10-19	Event identification
3	*prefer_loc*	varchar2(1)	a1	21-21	Preferred location identifier (`S`, `F`, `R`)
4	*constrain_ot*	number(1)	i1	23-23	Flag to constrain origin time
5	*constrain_latlon*	number(1)	i1	25-25	Flag to constrain latitude/longitude
6	*constrain_depth*	number(1)	i1	27-27	Flag to constrain depth
7	*src_dpnt_corr*	number(2)	i2	29-30	Source-dependent correction code
8	*loc_src_dpnt_reg*	varchar2(15)	a15	32-46	Region name of source-dependent location correction
9	*loc_sdv_screen*	number(1)	i1	48-48	Flag to ignore large data residuals in location
10	*loc_sdv_mult*	float(24)	f5.2	40-54	Location large residual multiplier factor
11	*loc_alpha_only*	number(1)	i1	56-56	Flag to use only primary stations in location
12	*loc_all_stas*	number(1)	i1	58-58	Flag to use only stations with *src_dpnt_corr*
13	*loc_dist_varwgt*	number(1)	i1	60-60	Flag to use distance variance weighting

Table A26. event_control (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
14	*mag_src_dpnt_reg*	varchar2(15)	a15	62-76	Region name of source-dependent magnitude correction
15	*mag_sdv_screen*	number(1)	i1	78-78	Flag to ignore large magnitude data residuals
16	*mag_sdv_mult*	float(24)	f5.2	80-84	Magnitude large residual multiplier factor
17	*mag_alpha_only*	number(1)	i1	86-86	Flag to limit station net used in magnitude
18	*mag_all_stas*	number(1)	i1	88-88	Flag to use only primary stations in magnitude
19	*mb_min_dist*	float(24)	f9.4	90-98	Minimum distance (degrees) for mb
20	*mb_max_dist*	float(24)	f9.4	100-108	Maximum distance (degrees) for mb
21	*mmodel*	varchar2(15)	a15	110-124	Network magnitude model
22	*cov_sm_axes*	float(24)	f9.4	126-134	Coverage ellipse semi-axes conversion factor
23	*cov_depth_time*	float(24)	f9.4	136-144	Coverage ellipse *depth/time* conversion factor
24	*lddate*	date	a19	146-164	Load date

Category:	Event Processing	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing	
Keys:	Primary	*evid/orid*
Data:	Descriptive	*prefer_loc, constrain_ot, constrain_latlon, constrain_depth, src_dpnt_corr, loc_src_dpnt_reg, loc_sdv_screen, loc_sdv_mult, loc_alpha_only, loc_all_stas, loc_dist_varwgt, mag_src_dpnt_reg, mag_sdv_screen, mag_sdv_mult, mag_alpha_only, mag_all_stas, mb_min_dist, mb_max_dist, mmodel cov_sm_axes, cov_depth_time*
	Administrative	*lddate*

ex_an

The **ex_an** table contains analyses of expert system solutions compared to analyst solutions from the *ExAnComp* application.

Table A27. ex_an

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*forid*	number(98)	i9	1-9	Final origin identifier
2	*eorid*	number(9)	i9	10-19	Expert system origin identifier
3	*ddist*	float(24)	f8.3	21-28	Distance between *forid* and *eorid*
4	*ddepth*	float(24)	f6.1	30-35	Depth difference
5	*dtime*	float(24)	f8.3	37-44	Origin time difference
6	*did*	varchar2(4)	a4	46-49	Identification difference
7	*dnsta*	number(8)	i8	51-58	Difference in recording stations
8	*dlsta*	number(8)	i8	60-67	Difference in local stations
9	*dasta*	number(8)	i8	69-76	Difference in regional array stations
10	*drsta*	number(8)	i8	78-85	Difference in non-array regional station
11	*dtsta*	number(8)	i8	87-94	Difference in teleseismic station
12	*dndef*	number(8)	i8	96-103	Difference in defining phases
13	*dprimp*	number(8)	i8	105-112	Difference in primary phases
14	*dsecondp*	number(8)	i8	114-121	Difference in secondary phases
15	*ddepthp*	number(8)	i8	123-130	Difference in depth phases
16	*rprimp*	number(8)	i8	132-139	Renamed primary phases
17	*rsecondp*	number(8)	i8	141-148	Renamed secondary phases
18	*rdepthp*	number(8)	i8	150-157	Renamed depth phases

Table A27. ex_an (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
19	*added*	number(8)	i8	159-166	Number of added phases
20	*retime*	number(8)	i8	168-175	Number of retimed phases
21	*splitev*	varchar2(4)	a4	177-180	Split event (y/n)
22	*multev*	varchar2(4)	a4	182-185	Multiple events (y/n)
23	*kbscause*	varchar2(7)	a7	187-193	Knowledge system explanation
24	*lddate*	date	a19	195-213	Load date

Category:	System Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*forid*
	Foreign	*eorid*
Data:	Measurement	*ddist, ddepth, dtime, did, dnsta, dlsta, dasta, drsta, dtsta, dndef, dprimp, dsecondp, ddepthp, rprimp, rsecondp, rdepthp, added, retime, splitev, multev, kbscause*
	Administrative	*lddate*

ftpfailed

The **ftpfailed** table facilitates ftp retrieval and the placement of data messages between contributing NDCs.

Table A28. ftpfailed

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*msgid*	number(9)	i9	1-9	Message identifier
2	*ftp_address*	varchar2(64)	a64	11-74	ftp address for auxilary data
3	*numfailedattempt*	number(2)	i4	76-79	Number of failed attempts
4	*lastfailedtime*	float(53)	f17.5	81-97	Time of most recent attempt
5	*ftpstatus*	varchar2(8)	a8	99-106	Status of ftp attempt (retry or failed)
6	*lddate*	date	a19	108-126	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*msgid*
	Foreign	*ftp_address*
Data:	Descriptive	*ftp_address, numfailedattempt, lastfailedtime, ftpstatus*
	Administrative	*lddate*

ftplogin

The **ftplogin** table is used by the auxiliary data retrieval system to obtain data via ftp from auxiliary stations.

Table A29. ftplogin

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*ftp_address*	varchar2(64)	a64	1-64	ftp address for auxiliary data
2	*username*	varchar2(24)	a24	66-89	User name for ftp access
3	*password*	varchar2(16)	a16	91-106	User password for ftp access
4	*lddate*	date	a19	108-126	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*ftp_address*
Data:	Descriptive	*username, password*
	Administrative	*lddate*

ga_tag

The **ga_tag** table contains information on the use of arrivals and origins in the *GA* application.

Table A30. ga_tag

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*objtype*	varchar2(1)	a1	1-1	Type of identifier (a for arrival, o for origin)
2	*id*	number(9)	i9	3-11	Identification number (*arid* or *orid*)
3	*process_state*	varchar2(20)	a20	13-32	Use of *arid* or *orid*

Category:	Network Processing	
CSCI(s)	Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*objtype/id/process_state*
Data:	Descriptive	*objtype*
	Measurement	*state*

gregion

The **gregion** table contains geographic region numbers and their equivalent descriptions.

Table A31. gregion

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*grn*	number(8)	i8	1-8	Geographic region number
2	*grname*	varchar2(40)	a40	10-49	Geographic region name
3	*lddate*	date	a19	51-69	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*grn*
Data:	Descriptive	*grn, grname*
	Administrative	*lddate*

hydro_arr_group

The **hydro_arr_group** table contains hydroacoustic arrival-based estimates of slowness and azimuth.

Table A32. hydro_arr_group

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*hydro_id*	number(9)	i9	1-9	Hydro-arrival-group identifier
2	*az1*	float(24)	f7.2	11-17	Azimuth estimated from lag times
3	*az2*	float(24)	f7.2	19-25	Second possible azimuth (2 arrivals)
4	*slow*	float(24)	f7.2	27-33	Slowness (s/km)
5	*delaz*	float(24)	f7.2	35-41	Azimuth uncertainty
6	*nhydarr*	number(4)	i4	43-46	Number of arrivals in hydro-arrival-group
7	*net*	varchar2(8)	a8	48-55	Hydro network name
8	*hydro_grp_phase*	varchar2(8)	a8	57-64	Hydro-arrival-group phase
9	*lddate*	date	a19	66-84	Load date

Category:	Hydro Azimuth Processing	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*hydro_id*
	Foreign	*net*
Data:	Descriptive	*nhydarr, hydro_grp_phase, slow*
	Measurement	*az1, az2, delaz*
	Administrative	*lddate*

hydro_arrival

The **hydro_arrival** table contains hydroacoustic arrival information such as duration and the crossing point lag of the signal, autocorrelation bubble pulse frequency, autocovariance peak ratio (*rt_ro*), cepstrum bubble pulse, bubble pulse amplitude versus root mean square (rms), filter ratio, normalized amplitude, sensor yield, and sensor yield error.

Table A33. hydro_arrival

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*arid*	number(9)	i9	1-9	Arrival identifier
2	*sta*	varchar2(6)	a6	11-16	Station
3	*duronset*	float(53)	f17.5	18-34	Duration onset time
4	*durend*	float(53)	f17.5	36-52	Duration end time
5	*onset_time*	float(53)	f17.5	54-70	Estimated onset time of signal
6	*termination_time*	float(53)	f17.5	72-88	Estimated termination time of signal
7	*cplag*	float(24)	f11.4	91-100	Crossing point lag of the signal
8	*bpfrqac*	float(24)	f11.4	102-112	Autocorrelation bubble pulse
9	*rt*	float(24)	f11.4	114-124	Autocovariance peak value
10	*bpfrqcep*	float(24)	f11.4	126-136	Cepstrum bubble pulse
11	*rms*	float(24)	f11.4	138-148	The rms amplitude from autocorrelation
12	*flt_rto*	float(24)	f11.4	151-160	Filter ratio
13	*normamp*	float(24)	f11.4	162-172	Normalized amplitude
14	*ampcorclip*	float(24)	f11.4	174-184	Correction to raw amplitude for clipping
15	*ampcordist*	float(24)	f11.4	186-196	Correction to raw amplitude for distance
16	*ampcordepth*	float(24)	f11.4	198-208	Correction to raw amp for depth
17	*yield*	float(24)	f11.4	210-220	Sensor yield

Table A33. hydro_arrival (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
18	*ylderr*	float(24)	f11.4	222-232	Sensor yield error
19	*commid*	number(9)	i9	234-242	Comment identifier
20	*lddate*	date	a19	244-262	Load date

Category:	Hydroacoustic Processing	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing	
Keys:	Primary	*arid*
	Foreign	*commid*
Data:	Descriptive	*sta*
	Measurement	*duronset, durend, onset_time, termination_time, cplag, bpfrqac, rt, bpfrqcep, rms, flt_rto, normamp, ampcorclip, ampcordist, ampcordepth, yield, ylderr*
	Administrative	*lddate*

hydro_assoc

The **hydro_assoc** table contains hydroacoustic arrival-based estimates of slowness and azimuth.

Table A34. hydro_assoc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*arid*	number(9)	i9	1-9	Arrival identifier
2	*hydro_id*	number(9)	i9	11-19	Hydro-arrival-group identifier
3	*azcontrib*	varchar2(1)	a1	21-21	Azimuth contribution flag (`y` or `n`)
4	*lddate*	date	a19	23-41	Load date

Category:	Hydro Azimuth Processing	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*arid*
	Foreign	*hydro_id*
Data:	Descriptive	*azcontrib*
	Administrative	*lddate*

hydro_origin

The **hydro_origin** table contains a summary of AFTAC-specific hydroacoustic origin information such as:

- Type of origin location, determined from:
  - Seismic system
  - Various unique hydroacoustic signal types, volcanic underwater, or undetermined
- Bubble pulse frequency used to calculate the yield value and that yield value error

This table also identifies if this origin is part of a series and the unique identifier for that series.

Table A35. hydro_origin

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identifier
2	*hydroloc_code*	number(8)	i8	11-18	Hydroacoustic origin location
3	*hydroyield*	float(24)	f11.2	20-30	Hydroacoustic event yield
4	*hydroylderr*	float(24)	f11.2	32-42	Hydroacoustic event yield error
5	*num_in_series*	number(8)	i8	44-51	Number of event in series
6	*serid*	number(8)	i8	53-60	Series identifier
7	*hyd_class_code*	number(8)	i8	62-69	Hydroacoustic event classification
8	*commid*	number(9)	i9	71-79	Comment identifier
9	*lddate*	date	a19	81-98	Load date

Category:	Hydroacoustic Processing	
CSCI(s)	Data Management, Automatic Processing	
Keys:	Primary	*orid*
	Foreign	*commid*
Data:	Descriptive	*hydroloc_code, num_in_series, serid, hyd_class_code*
	Measurement	*hydroyield, hydroylderr*
	Administrative	*lddate*

instrument

The **instrument** table contains ancillary calibration information. It holds nominal one-frequency calibration factors for each instrument and pointers to nominal frequency-dependent calibration for an instrument. This table also holds pointers to the exact calibrations obtained by direct measurement on a particular instrument (see **sensor**).

Table A36. instrument

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*inid*	number(8)	i8	1-8	Instrument identifier
2	*insname*	varchar2(50)	a50	10-59	Instrument name
3	*instype*	varchar2(6)	a6	61-66	Instrument type
4	*band*	varchar2(1)	a1	68-68	Frequency band
5	*digital*	varchar2(1)	a1	70-70	Data type, digital (d), or analog (a)
6	*samprate*	float(24)	f11.7	72-82	Sampling rate in samples/second
7	*ncalib*	float(24)	f16.6	84-99	Nominal calibration (nanometers/digital count)
8	*ncalper*	float(24)	f16.6	101-116	Nominal calibration period (seconds)
9	*dir*	varchar2(64)	a64	118-181	Directory
10	*dfile*	varchar2(32)	a32	183-214	Data file
11	*rsptype*	varchar2(6)	a6	216-221	Response type
12	*lddate*	date	a19	223-241	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*inid*
Data:	Descriptive	*insname, instype, band, digital, dir, dfile, rsptype*
	Measurement	*samprate, ncalib, ncalper*
	Administrative	*lddate*

interval

The **interval** table defines units of processing. The *time, endtime,* and *name* types indicate processing times for a named object. The *class* type allows a single **interval** table to be used for different classes of objects.

Table A37. interval

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*intvlid*	number(9)	i9	1-9	Interval identifier
2	*class*	varchar2(16)	a16	11-26	Type of interval
3	*name*	varchar2(20)	a20	28-47	Name of interval
4	*time*	float(53)	f17.5	49-65	Starting time of data
5	*endtime*	float(53)	f17.5	67-83	Ending time of data
6	*state*	varchar2(16)	a16	85-100	Current processing state
7	*moddate*	date	a17	102-118	Time of last processing state change
8	*auth*	varchar2(15)	a15	120-134	Author of interval
9	*lddate*	date	a19	136-154	Load date

Category:	Distributed Processing	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*class/name/time/endtime*
	Alternate	*intvlid*
Data:	Descriptive	*class, name, state*
	Measurement	*time, endtime*
	Administrative	*auth, moddate, lddate*

interval_files

The **interval_files** table contains a description of a file placed in the archive.

Table A38. interval_files

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*intvlid*	number(9)	i9	1-9	Interval identifier
2	*ftype*	varchar2(2)	a1	11-11	Archive file type [archive (`a`), waveform (`w`), directory (`d`)]
3	*subtype*	varchar2(20)	a20	13-32	Subtype of the given *ftype*
4	*location*	varchar2(20)	a20	34-53	Location code for the file
5	*dir*	varchar2(64)	a64	55-118	Directory
6	*dfile*	varchar2(32)	a32	120-151	Data file
7	*lddate*	date	a19	153-171	Load date

Category:	Data Archiving	
CSCI(s)	Data Management	
Keys:	Primary	*intvlid*
Data:	Descriptive	*ftype, subtype, location, dir, dfile*
	Administrative	*lddate*

lastid

The **lastid** table contains counter values (the last value used for keys). This table is a reference table from which programs may retrieve the last sequential value of one of the numeric keys. Unique keys are required before inserting a record in numerous tables. The **lastid** table has exactly one row for each *keyname*.

Table A39. lastid

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*keyname*	varchar2(15)	a15	1-15	Identifier name (*arid, orid,* etc.)
2	*keyvalue*	number(9)	i9	17-25	Last value used for that identifier
3	*lddate*	date	a19	28-46	Load date

Category:	Data Administration	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*keyname*
Data:	Descriptive	*keyname, keyvalue*
	Administrative	*lddate*

mapcolor

The **mapcolor** table contains information to associate a *mapid* from the **mapdisc** table with a *colormapid* from the **colordisc** table. This table is used to plot the same map (*mapid*) in different colors (for example, brown, green, or outline).

Table A40. mapcolor

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*mapid*	number(8)	i8	1-8	Map identifier
2	*colormapid*	number(8)	i8	10-17	Colormap identifier
3	*lddate*	date	a19	19-37	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*mapid/colormapid*
Data:	Administrative	*lddate*

mapdisc

The **mapdisc** table contains information about map files that are on disk.

Table A41. mapdisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*mapid*	number(8)	i8	1-8	Map identifier
2	*mapname*	varchar2(64)	a64	10-73	Map name
3	*dfile*	varchar2(32)	a32	75-106	Map data filename
4	*dir*	varchar2(64)	a64	108-171	Directory
5	*maptype*	number(8)	i8	173-180	Map type
6	*mapfiletype*	varchar2(4)	a4	182-185	Map file type
7	*projection*	number(8)	i8	187-194	Map projection
8	*dimx*	number(8)	i8	196-203	Map x dimension
9	*dimy*	number(8)	i8	205-212	Map y dimension
10	*reflon*	float(53)	f11.6	214-224	Reference longitude
11	*reflat*	float(53)	f11.6	226-236	Reference latitude
12	*refoffsetlon*	float(24)	f9.4	238-246	Longitude reference offset
13	*refoffsetlat*	float(24)	f9.4	248-256	Latitude reference offset
14	*lonorigradians*	float(24)	f9.4	258-266	Longitude origin radians
15	*latorigradians*	float(24)	f9.4	268-276	Latitude origin radians
16	*scale*	float(24)	f9.4	278-286	Map scale
17	*rotation*	float(24)	f9.4	288-296	Map rotation
18	*latminor*	float(53)	f11.6	298-308	Latitude interval for minor grid lines
19	*latmajor*	float(53)	f11.6	310-320	Latitude interval for major grid lines
20	*lonminor*	float(53)	f11.6	322-332	Longitude interval for minor grid lines
21	*lonmajor*	float(53)	f11.6	334-344	Longitude interval for major grid lines
22	*bordercolor*	varchar2(32)	a32	346-377	Border color name

Table A41. mapdisc (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
23	*label*	varchar2(65)	a65	379-443	Map category
24 to 38	*gctp1 through gctp15*	float(24)	f10.4	445-608	General cartographic transformation package variables
39	*lddate*	date	a19	610-628	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*mapid*
Data:	Descriptive	*mapname, dfile, dir, maptype, mapfiletype, projection, bordercolor, label dimx, dimy, reflon,*
	Measurement	*reflat, refoffsetlon, refoffsetlat, lonorigradians, latorigradians, scale, rotation, latminor, latmajor, lonminor, lonmajor, gctp1–15*
	Administrative	*lddate*

mapover

The **mapover** table contains links between the **mapdisc** and **overlaydisc** tables.

Table A42. mapover

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*mapid*	number(8)	i8	1-8	Map identifier
2	*overlayid*	number(8)	i8	10-17	Overlay identifier
3	*lddate*	date	a19	19-37	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*mapid/overlayid*
Data:	Administrative	*lddate*

mappoint

The **mappoint** table contains labeled point data to be displayed by the *Map* application.

Table A43. mappoint

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*lat*	float(53)	f111.6	1-11	Latitude
2	*lon*	float(53)	f11.6	13-23	Longitude
3	*mplabel*	varchar2(65)	a65	25-89	Map point label
4	*mptype*	varchar2(20)	a20	91-110	Map point type
5	*mpdescrip*	varchar2(50)	a50	112-161	Map point description
6	*lddate*	date	a19	163-181	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*lat/lon/mptype*
Data:	Descriptive	*lat, lon, mplabel, mptype, mpdescrip*
	Administrative	*lddate*

mig_date

The **mig_date** table is used by the *MigrateData* application to track table migration based on *lddate* as opposed to time interval as in the **timestamp** table.

Table A44. mig_date

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*procclass*	varchar2(16)	a16	1-16	Process class
2	*procname*	varchar2(16)	a16	18-33	Process name
3	*last_mig_date*	date	a17	35-51	Last migration date
4	*lddate*	date	a19	53-71	Load date

Category:	Data Migration	
CSCI(s)	Data Management	
Keys:	Primary	*procclass/procname*
Data:	Descriptive	*procclass, procname, last_mig_date*
	Administrative	*lddate*

mig_rules

The **mig_rules** table contains rules for migrating database tables from one database table to another.

Table A45. mig_rules

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*task_num*	number(4)	i4	1-4	Order of this task
2	*query_type*	varchar2(20)	a20	6-25	Type of query
3	*src*	varchar2(10)	a10	27-36	Source database
4	*src_tbl*	varchar2(30)	a30	38-67	Source table
5	*dest*	varchar2(10)	a10	69-78	Destination database
6	*dest_tbl*	varchar2(30)	a30	80-109	Destination table
7	*quer_seq_no*	number(4)	i4	111-114	Order of this part of the query
8	*seq_type*	varchar2(15)	a15	116-130	Type of sequence to be added to the query
9	*seq_contents*	varchar2(200)	a200	132-331	Query contents

Category:	Data Migration	
CSCI(s)	Data Management	
Keys:	Primary	*task_num/query_type/src_tbl*
Data:	Descriptive	*task_num, query_type, src, src_tbl, dest, dest_tbl, quer_seq_no, seq_type, seq_contents*

missed_class

The **missed_class** table contains information pertaining to events identified by only one bulletin during a bulletin comparison.

Table A46. missed_class

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-0	Origin identifier
2	*nsta*	number(8)	i8	11-18	Number of stations used
3	*telep*	number(8)	i8	20-27	Indicates the number of teleseismic p phases in an event
4	*sigdet*	number(8)	i8	29-36	Indicates number of arrivals detected in both bulletins
5	*assoc*	number(8)	i8	38-45	Indicates number of associated detections in second bulletin not detected in the first
6	*bulletins*	varchar2(64)	a64	47-110	Description
7	*time*	float(53)	f17.5	112-128	Start time of missed class
8	*endtime*	float(53)	f17.5	130-146	End time of missed class
9	*lddate*	date	a19	148-166	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*orid*
Data:	Descriptive	*bulletins*
	Measurement	*nsta, telep, sigdet, assoc, time, endtime*
	Administrative	*lddate*

msgaux

The **msgaux** table contains records of unsuccessfully processed Automatic Data Request Manager (*AutoDRM)* messages.

Table A47. msgaux

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*msgid*	number(9)	i9	1-9	Message identifier
2	*msgrow*	number(4)	i4	11-14	Line number in message
3	*statecount*	number(8)	i8	16-23	Number of failures
4	*command*	varchar2(24)	a24	25-48	Command that could not be processed
5	*sub_status*	varchar2(24)	a24	50-73	Cause of failure
6	*lddate*	date	a19	75-93	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*msgid/msgrow/statecount*
Data:	Descriptive	*command, sub_status*
	Measurement	*msgrow, statecount*
	Administrative	*lddate*

msgdatatype

The **msgdatatype** table supports data tracking by recording each data section in a message for incoming and outgoing messages.

Table A48. msgdatatype

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*msgid*	number(9)	i9	1-9	Message identifier
2	*msgdtype*	varchar2(16)	a16	11-26	Data type of the data section within the message
3	*msgdformat*	varchar2(16)	a16	28-43	General format of data that follows
4	*msgstatus*	varchar2(32)	a32	45-76	Status of the data section
5	*foff*	number(10)	i10	78-87	File offset to beginning of data section
6	*msize*	number(8)	i8	89-96	Size of data section
7	*lddate*	date	a19	98-116	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*msgid/foff*
Data:	Descriptive	*msgdtype, msgdformat*
	Measurement	*msgstatus, foff, msize*
	Administrative	*lddate*

msgdest

The **msgdest** table contains information about messages sent.

Table A49. msgdest

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*msgdid*	number(9)	i9	1-9	Message destination identifier
2	*msgid*	number(9)	i9	11-19	Message identifier
3	*transmeth*	varchar2(16)	a16	21-36	Method by which the response is to be delivered to the requester
4	*emailto*	varchar2(64)	a64	38-101	E-mail address to send message
5	*msgstatus*	varchar2(32)	a32	103-134	Current status of the response message
6	*itime*	float(53)	f17.5	136-152	Time at which table entry was made
7	*timesent*	float(53)	fl17.5	154-170	Time at which message was sent
8	*lddate*	date	a19	172-190	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*msgdid*
	Foreign	*msgid*
Data:	Descriptive	*transmeth*
	Measurement	*emailto, msgstatus, timesent, itime*
	Administrative	*lddate*

msgdisc

The **msgdisc** table contains information pertinent to messages including the date and time that the message was sent or received, identification information, and where the message is stored.

Table A50. msgdisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*msgid*	number(9)	i9	1-9	Message identifier
2	*userid*	number(8)	i8	11-18	User identifier
3	*msgver*	varchar2(8)	a8	21-27	Message system version number
4	*msgtype*	varchar2(16)	a16	29-44	Message type
5	*subtype*	varchar2(20)	a20	46-65	Message subtype
6	*extmsgid*	varchar2(20)	a20	67-86	Message identification string provided by the sender
7	*intid*	number(9)	i9	88-96	Either the locally generated *msgid* of an earlier table entry that evoked the creation of this table entry or the *reqid* from the **request** table of an internally generated request
8	*intidtype*	varchar2(16)	a16	98-103	*Intid* type
9	*msgsrc*	varchar2(16)	a16	105-120	Message source code
10	*itime*	float(53)	f17.5	122-138	Initial time message was received
11	*idate*	number(8)	i8	140-147	Initial date message was received
12	*imethod*	varchar2(8)	a8	149-156	Input method (e-mail or ftp)
13	*isrc*	varchar2(64)	a64	158-221	Initial source of message
14	*msize*	number(8)	i8	223-230	Message size in bytes
15	*msgstatus*	varchar2(32)	a32	232-263	Status of message
16	*subject*	varchar2(64)	a64	265-328	Subject header from e-mail message

Table A50. msgdisc (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
17	*dir*	varchar2(64)	a64	330-395	Directory to find file
18	*dfile*	varchar2(32)	a32	395-426	Name of data file
19	*foff*	number(10)	i10	428-437	Byte offset of data segment within file
20	*mfoff*	number(8)	i8	439-446	Offset in bytes to beginning of message
21	*fileoff*	number(8)	i8	448-455	Number of bytes to the first character of the e-mail file (first character of the e-mail header)
22	*filesize*	number(8)	i8	457-464	Size of file
23	*sigtype*	varchar2(64)	a64	466-529	Digital signature type
24	*verifstatus*	varchar2(4)	a4	531-534	Status of verification
25	*commid*	number(8)	i8	536-543	Comment identifier
26	*lddate*	date	a19	545-563	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*msgid*
	Foreign	*userid, intid, commid*
Data:	Descriptive	*msgver, msgtype, intidtype, subtype, msgsrc, msgstatus, subject, dir, dfile, foff, mfoff, fileoff, filesize, sigtype, verifstatus*
	Measurement	*extmsgid, intid, itime, idate, imethod, isrc, msize*
	Administrative	*lddate*

netmag

The **netmag** table contain estimates of network magnitudes of different types for an event. Each network magnitude has a unique *magid*. Station magnitudes used to compute the network magnitude are in the **stamag** table.

Table A51. netmag

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*magid*	number(9)	i9	1-9	Network magnitude identifier
2	*net*	varchar2(8)	a8	11-18	Unique network identifier
3	*orid*	number(9)	i9	20-28	Origin identifier
4	*evid*	number(9)	i9	30-38	Event identifier
5	*magtype*	varchar2(6)	a6	40-45	Magnitude type (`ms`, `mb`, etc.)
6	*nsta*	number(8)	i8	47-54	Number of stations used
7	*magnitude*	float(24)	f7.2	56-62	Magnitude
8	*uncertainty*	float(24)	f7.2	64-70	Magnitude uncertainty
9	*auth*	varchar2(15)	a15	72-86	Source/originator
10	*commid*	number(9)	i9	88-96	Comment identifier
11	*lddate*	date	a19	98-116	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*magid*
	Foreign	*evid, net, orid, commid*
Data:	Descriptive	*net, magtype*
	Measurement	*magnitude, nsta, uncertainty*
	Administrative	*auth, lddate*

network

The **network** table contains general information about seismic networks (see **affiliation**).

Table A52. network

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*net*	varchar2(8)	a8	1-8	Unique network identifier
2	*netname*	varchar2(80)	a80	10-89	Network name
3	*nettype*	varchar2(4)	a4	91-94	Network type (array, local, world-wide, etc.)
4	*auth*	varchar2(15)	a15	96-110	Source/originator
5	*commid*	number(9)	i9	112-120	Comment identifier
6	*lddate*	date	a19	122-140	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*net*
	Foreign	*commid*
Data:	Descriptive	*net, netname, nettype*
	Administrative	*auth, lddate*

origerr

The **origerr** table contains summaries of confidence bounds in origin estimations. The **origerr_ga** table is used by the GA application to store temporary origin error information. The measurement types are the elements of the location covariance matrix. The descriptive types give the uncertainties in location, depth, and origin time. These quantities are calculated from the covariance matrix, assuming gaussian errors and a confidence level *conf*.

Table A53. origerr

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*orid*	number(9)	i9	1-9	Origin identifier
2 to 11	*sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz*	float(24)	f15.4	11-169	Covariance matrix elements
12	*sdobs*	float(24)	f9.4	171-179	Standard error of observations
13	*smajax*	float(24)	f9.4	181-189	Semi-major axis of error
14	*sminax*	float(24)	f9.4	191-199	Semi-minor axis of error
15	*strike*	float(24)	f6.2	201-206	Strike of the semi-major axis
16	*sdepth*	float(24)	f9.4	208-216	Depth error
17	*stime*	float(24)	f6.3	218-223	Origin time error
18	*conf*	float(24)	f5.3	225-229	Confidence
19	*commid*	number(9)	i9	231-239	Comment identifier
20	*lddate*	date	a19	241-259	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*orid*
	Foreign	*commid*
Data:	Descriptive	*sdobs, smajax, sminax, strike, sdepth, stime, conf*
	Measurement	*sxx, syy, szz, stt, sxy, sxz, syz, stx, sty, stz*
	Administrative	*lddate*

origin (origin_ga)

The **origin** and **origin_ga** tables contain information describing a derived or reported origin for a particular event.

Table A54. origin (origin_ga)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*lat*	float(24)	f11.4	1-11	Estimated latitude
2	*lon*	float(24)	f11.4	13-23	Estimated longitude
3	*depth*	float(24)	f9.4	25-33	Estimated depth
4	*time*	float(53)	f17.5	35-51	Epoch time
5	*orid*	number(9)	i9	53-61	Origin identifier
6	*evid*	number(9)	i9	63-71	Event identifier
7	*jdate*	number(8)	i8	73-80	Julian date
8	*nass*	number(4)	i4	82-85	Number of associated phases
9	*ndef*	number(4)	i4	87-90	Number of locating phases
10	*ndp*	number(4)	i4	92-95	Number of depth phases
11	*grn*	number(8)	i8	97-104	Geographic region number
12	*srn*	number(8)	i8	106-113	Seismic region number
13	*etype*	varchar2(7)	a7	115-121	Event type
14	*depdp*	float(24)	f9.4	123-131	Estimated depth from depth phases
15	*dtype*	varchar2(1)	a1	133-133	Depth method used
16	*mb*	float(24)	f7.2	135-141	Body wave magnitude
17	*mbid*	number(9)	i9	143-151	M_b magnitude identifier
18	*ms*	float(24)	f7.2	153-159	Surface wave magnitude
19	*msid*	number(9)	i9	161-169	M_s magnitude identifier
20	*ml*	float(24)	f7.2	171-177	Local magnitude
21	*mlid*	number(9)	i9	179-187	M_L magnitude identifier

Table A54. origin (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
22	*algorithm*	varchar2(15)	a15	189-203	Location algorithm used
23	*auth*	varchar2(15)	a15	205-219	Source/originator
24	*commid*	number(9)	i9	221-229	Comment identifier
25	*lddate*	date	a19	231-249	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*orid*
	Alternate	*lat/lon/depth/time*
	Foreign	*evid, mbid, msid, mlid, commid*
Data:	Descriptive	*nass, ndef, ndp, grn, srn, etype*
	Measurement	*lat, lon, depth, time, jdate, depdp, dtype, mb, mbid, ms, msid, ml, mild*
	Administrative	*algorithm, auth, lddate*

overlaydisc

The **overlaydisc** table contains the location of the overlays for the *Map* application.

Table A55. overlaydisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*overlayid*	number(8)	i8	1-8	Overlay identifier
2	*overlayname*	varchar2(64)	a64	10-73	Overlay name
3	*dfile*	varchar2(32)	a32	75-106	Data filename
4	*dir*	varchar2(64)	a64	108-171	Directory
5	*colorname*	varchar2(32)	a32	173-204	Overlay color name
6	*lddate*	date	a19	206-224	Load date

Category:	Map	
CSCI(s)	Data Management, Interactive Processing	
Keys:	Primary	*overlayid*
Data:	Descriptive	*overlayname, dfile, dir, colorname*
	Administrative	*lddate*

pixdisc

The **pixdisc** table records images generated for *PerfMon*.

Table A56. pixdisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*pixid*	number(9)	i9	1-9	Picture identifier
2	*time*	float(53)	f17.5	11-27	Epoch time of start of recording period
3	*endtime*	float(53)	f17.5	29-45	Epoch time of end of recording period
4	*jdate*	number(8)	i8	47-54	Julian date
5	*topic*	varchar2(8)	a8	56-63	Type of data processing
6	*subtopic*	varchar2(8)	a8	65-72	Subtype of data processing
7	*processing*	varchar2(16)	a16	74-89	*PerfMon* mode
8	*grname*	varchar2(40)	a40	91-130	Image base map name
9	*pixdescr*	varchar2(64)	a64	132-195	Description of image
10	*pub_access*	number(8)	i8	197-204	Access permissions to images
11	*dir*	varchar2(64)	a64	206-269	Directory
12	*dfile*	varchar2(32)	a32	271-302	Data file name
13	*auth*	varchar2(15)	a15	304-318	Author
14	*lddate*	date	a19	320-338	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*pixid*
Data:	Descriptive	*time, endtime, jdate, topic, subtopic, pixdescr, dir, dfile, processing, grname, pub_access*
	Administrative	*auth, lddate*

qcdata

The **qcdata** table contains performance monitoring data quality information.

Table A57. qcdata

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*qcdataid*	number(9)	i9	1-9	QC data identifier
2	*sta*	varchar2(6)	a6	11-16	Station
3	*time*	float(53)	f17.5	18-34	Epoch start time of recording period
4	*endtime*	float(53)	f17.5	36-52	Epoch end time of recording period
5	*jdate*	number(8)	i8	54-61	Julian date
6	*nchans*	number(8)	i8	63-70	Number of channels
7	*expected*	float(53)	f12.1	72-83	Expected number of seconds of data
8	*retrieved*	float(24)	f12.1	85-96	Actual number of seconds of data
9	*masked*	float(53)	f17.5	98-114	Number of seconds masked out
10	*masks*	number(8)	i8	115-123	Number of masks
11	*noise*	float(24)	f8.3	125-132	Average noise amplitude
12	*auth*	varchar2(15)	a15	134-148	Author
13	*commid*	number(9)	i9	150-158	Comment identifier
14	*lddate*	date	a19	160-178	Load date

Category:	Performance Monitoring	
CSCI(s)	Data Management, Performance Monitoring	
Keys:	Primary	*qcdataid*
	Foreign	*commid*
Data:	Descriptive	*sta, time, endtime, jdate, nchans, expected, retrieved, masked, masks, noise*
	Administrative	*auth, lddate*

qcstats

The **qcstats** table contains waveform data quality statistics.

Table A58. qcstats

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*qcstatsid*	number(9)	i9	1-9	Data quality statistics identifier
2	*sta*	varchar2(6)	a6	11-16	Station name
3	*chan*	varchar2(8)	a8	18-25	Channel name
4	*time*	float(53)	f17.5	27-43	Interval start time
5	*jdate*	number(8)	i8	45-52	Julian date
6	*endtime*	float(53)	f17.5	54-70	Interval end time
7	*dettime*	float(53)	f17.5	72-88	Detection interval start time
8	*detendtime*	float(53)	f17.5	90-106	Detection interval end time
9	*missing*	float(53)	f17.5	108-124	Number of seconds of missing data
10	*dropped*	number(8)	i8	126-133	Flag indicating if interval was dropped
11	*nseg*	number(8)	i8	135-142	Number of masked segments
12	*masked*	float(53)	f17.5	145-160	Number of seconds in masked segments
13	*pointspike*	float(53)	f17.5	162-178	Number of seconds in masked segments due to point-spikes
14	*spike*	float(53)	f17.5	180-196	Number of seconds in masked segments due to spikes
15	*nconstseg*	number(8)	i8	198-205	Number of constant valued segments
16	*const*	float(53)	f17.5	207-223	Number of seconds masked due to constant valued segments

Table A58. qcstats (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
17	*avgconstval*	float(53)	f17.5	225-241	Average number of seconds in constant segments
18	*stdconstval*	float(53)	f17.5	243-259	Standard deviation of constant values
19	*auth*	varchar2(15)	a15	261-275	Author
20	*lddate*	date	a19	277-295	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring	
Keys:	Primary	*qcstatsid*
	Foreign	*sta, chan, time, endtime*
Data:	Descriptive	*sta, chan*
	Measurement	*time, jdate, endtime, dettime, detendtime, missing, dropped, nseg, masked, pointspike, spike, nconstseg, const, avgconstval, stdconstval*
	Administrative	*auth, lddate*

remark

The **remark** table contains comments. This table may be used to store free-form comments that embellish records of other tables. The *commid* type in many tables refers to a record in the **remark** table. If *commid* is NA (`-1`) in a record of any other table, no comments are stored for that record.

Table A59. remark

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*commid*	number(9)	i9	1-9	Comment identifier
2	*lineno*	number(8)	i8	11-18	Comment line number
3	*remark*	varchar2(80)	a80	20-99	Free-format comment
4	*lddate*	date	a19	101-119	Load date

Category:	Data Administration	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*commid/lineno*
Data:	Descriptive	*lineno, remark*
	Administrative	*lddate*

request

The **request** table defines segments of auxiliary waveform data to be acquired. The *time*, *endtime, sta,* and *chan* types define a single unit of data. Data import programs must succeed in acquiring all the data for a time interval before changing the state to indicate success.

Table A60. request

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*reqid*	number(9)	i9	1-9	Request identifier
2	*sta*	varchar2(6)	a6	11-16	Station code
3	*chan*	varchar2(8)	a8	18-25	Channel code
4	*array*	varchar2(8)	a8	27-34	Array code
5	*orid*	number(9)	i9	36-44	Origin identifier
6	*evid*	number(9)	i9	46-54	Event identifier
7	*time*	float(53)	f17.5	56-72	Starting time of requested waveform data
8	*endtime*	float(53)	f17.5	74-90	Ending time of requested waveform data
9	*class*	varchar2(16)	a16	92-107	Type of request
10	*req_state*	varchar2(16)	a16	109-124	Current request state
11	*statecount*	number(8)	i8	126-133	Number of failed attempts (when state = `failed`)
12	*complete*	number(8)	i8	135-142	Percentage of data acquired
13	*requestor*	varchar2(15)	a15	144-158	Original author of record

Table A60. request (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
14	*modtime*	float(53)	f17.5	160-176	Time of last state change (epoch time)
15	*modauthor*	varchar2(15)	a15	178-192	Author of last state change
16	*lddate*	date	a19	194-212	Load date

Category:	Message Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*reqid*
	Alternate	*sta/chan/ time/endtime*
	Foreign	*orid, evid*
Data:	Descriptive	*sta, chan, array, class requestor, modauthor*
	Measurement	*time, endtime, reqstate, statecount, complete, modtime*
	Administrative	*lddate*

sensor

The **sensor** table contains calibration information for specific sensor channels. This table provides a record of updates in the calibration factor or clock error of each instrument and links a *sta/chan/time* to a complete instrument response in the **instrument** table. Waveform data are converted into physical units through multiplication by the *calib* type located in **wfdisc**. The correct value of *calib* may not be accurately known when the **wfdisc** record is entered into the database. The **sensor** table provides the mechanism (*calratio* and *calper*) to update *calib*, without requiring possibly hundreds of **wfdisc** records to be updated. Through the foreign key *inid*, this table is linked to **instrument**, which has types pointing to flat files holding detailed calibration information in a variety of formats (see **instrument**).

Table A61. sensor

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*time*	float(53)	f17.5	17-33	Epoch time of start of recording period
4	*endtime*	float(53)	f17.5	35-51	Epoch time of end of recording period
5	*inid*	number(8)	I8	53-60	Instrument identifier
6	*chanid*	number(8)	I8	62-69	Channel identifier
7	*jdate*	number(8)	I8	71-78	Julian date
8	*calratio*	float(24)	f16.6	80-95	Calibration
9	*calper*	float(24)	f16.6	97-112	Calibration period
10	*tshift*	float(24)	f16.2	114-129	Correction of data processing time

Table A61. sensor (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
11	*instant*	varchar2(1)	a1	131-131	Discrete/continuing (`y`, `n`) snapshot
12	*lddate*	date	a19	133-151	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/chan/time/endtime*
	Foreign	*inid, chanid*
Data:	Descriptive	*sta, chan, instant*
	Measurement	*time, endtime, jdate, calratio, calper, tshift*
	Administrative	*lddate*

site

The **site** table contains station location information. It names and describes a point on the earth where measurements are made (for example, the location of an instrument or array of instruments). This table contains information that normally changes infrequently, such as location. In addition, the **site** table contains types that describe the offset of a station relative to an array reference location. Global data integrity implies that the *sta*/*ondate* in **site** be consistent with the *sta*/*chan*/*ondate* in the **sitechan** table.

Table A62. site

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station identifier
2	*ondate*	number(8)	i8	8-15	Julian start date
3	*offdate*	number(8)	i8	17-24	Julian off date
4	*lat*	float(53)	f11.6	26-36	Latitude
5	*lon*	float(53)	f11.6	38-48	Longitude
6	*elev*	float(24)	f9.4	50-58	Elevation
7	*staname*	varchar2(50)	a50	60-109	Station description
8	*statype*	varchar2(4)	a4	111-114	Station type (single station, array)
9	*refsta*	varchar2(6)	a6	116-121	Reference station for array members
10	*dnorth*	float(24)	f9.4	123-131	Offset from array reference (km)
11	*deast*	float(24)	f9.4	133-141	Offset from array reference (km)
12	*lddate*	date	a19	143-161	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/ondate*
Data:	Descriptive	*sta, staname, statype, refsta*
	Measurement	*ondate, offdate, lat, lon, elev, dnorth, deast*
	Administrative	*lddate*

siteaux

The **siteaux** table contains additional site-dependent parameters that are not included in the **site** table.

Table A63. siteaux

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*time*	float(53)	f17.5	17-33	Epoch time
4	*nois*	float(24)	f10.1	35-44	Noise amplitude
5	*noissd*	float(24)	f5.2	46-50	Standard deviation of log noise
6	*amcor*	float(24)	f10.1	52-61	Amplitude correction
7	*amcorsd*	float(24)	f5.2	63-67	Correction standard deviation
8	*snthrsh*	float(24)	f5.2	69-73	Signal/noise detection threshold
9	*rely*	float(24)	f5.2	75-79	Station reliability
10	*ptmcor*	float(24)	f6.3	81-86	P arrival time correction
11	*stmcor*	float(24)	f6.3	88-93	S arrival time correction
12	*staper*	float(24)	f5.2	95-99	Period for measurements
13	*auth*	varchar2(15)	a15	101-115	Author
14	*commid*	number(98)	i9	117-125	Comment identifier
15	*lddate*	date	a19	127-145	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/chan/time*
	Foreign	*commid*
Data:	Descriptive	*sta, chan*
	Measurement	*time, nois, noissd, amcor, amcorsd, snthrsh, rely, ptmcor, stmcor, staper*
	Administrative	*auth, lddate*

sitechan

The **sitechan** table contains station-channel information. It describes the orientation of a recording channel at the site referenced by *sta*. The table provides information about the various channels that are available at a station and maintains a record of the physical channel configuration at a site.

Table A64. sitechan

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station identifier
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*ondate*	number(8)	i8	17-24	Julian start date
4	*chanid*	number(8)	i8	26-33	Channel identifier
5	*offdate*	number(8)	i8	35-42	Julian off date
6	*ctype*	varchar2(4)	a4	44-47	Channel type
7	*edepth*	float(24)	f9.4	49-57	Emplacement depth
8	*hang*	float(24)	f6.1	59-64	Horizontal angle
9	*vang*	float(24)	f6.1	66-71	Vertical angle
10	*descrip*	varchar2(50)	a50	73-122	Channel description
11	*lddate*	date	a19	124-142	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta/chan/ondate*
	Alternate	*chanid*
Data:	Descriptive	*sta, chan, ctype, descrip*
	Measurement	*ondate, offdate, edepth, hang, vang*
	Administrative	*lddate*

sregion

The **sregion** table contains seismic region numbers and their equivalent descriptions.

Table A65. sregion

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
	srn	number(8)	i8	1-8	Seismic region number
	srname	varchar2(40)	a40	10-49	Seismic region name
	lddate	date	a19	51-69	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Management	
Keys:	Primary	*srn*
Data:	Descriptive	*srn, srname*
	Administrative	*auth, lddate*

stamag

The **stamag** table contain station magnitude estimates based upon measurements made on specific seismic phases. Values in the **stamag** table are used to calculate network magnitudes stored in the **netmag** table.

Table A66. stamag

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*magid*	number(9)	i9	1-9	Magnitude identifier
2	*ampid*	number(9)	i9	11-19	Amplitude identifier
3	*sta*	varchar2(6)	a6	21-26	Station code
4	*arid*	number(9)	i9	28-36	Arrival identifier
5	*orid*	number(9)	i9	38-46	Origin identifier
6	*evid*	number(9)	i9	48-56	Event identifier
7	*phase*	varchar2(8)	a8	58-65	Associated phase
8	*delta*	float(24)	f8.3	67-74	Station-to-event distance
9	*magtype*	varchar2(6)	a6	76-81	Magnitude type (`ml`, `ms`, `mb`, etc.)
10	*magnitude*	float(24)	f7.2	83-89	Magnitude
11	*uncertainty*	float(24)	f7.2	91-97	Magnitude uncertainty
12	*magres*	float(24)	f7.2	99-105	Magnitude residual
13	*magdef*	varchar2(1)	a1	107-107	`d` or `n` flag indicating if magnitude is defining or nondefining
14	*mmodel*	varchar2(15)	a15	109-123	Magnitude model
15	*auth*	varchar2(15)	a15	125-139	Author
16	*commid*	number(9)	i9	141-149	Comment identifier
17	*lddate*	date	a19	151-169	Load date

Category:	Fundamental Reference	
CSCI(s)	Data Management, Automatic Processing, Interactive Processing, Performance Monitoring, Tuning Tools	
Keys:	Primary	*magid/ampid/sta*
	Foreign	*arid, orid, evid, commid*
Data:	Descriptive	*delta, sta, phase, magtype, magdef, mmodel*
	Measurement	*magnitude, uncertainty, magres*
	Administrative	*auth, lddate*

station_hist

The **station_hist** table contains performance monitoring station processing history.

Table A67. station_hist

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station
2	*pmdescr*	varchar2(64)	a64	8-71	Description of *PerfMon* state
3	*detections*	float(24)	f10.2	73-82	Number of arrivals normalized to one day
4	*assoc_dets*	float(24)	f11.2	84-94	Normalized number of associated detections
5	*added_dets*	float(24)	f10.2	96-105	Normalized number of added detections
6	*dets_az*	float(24)	f10.2	107-116	Normalized number of detections affecting *mean_az*
7	*dets_slo*	float(24)	f10.2	118-127	Normalized number of detections affecting *mean_slo*
8	*dets_time*	float(24)	f10.2	129-138	Normalized number of detections affecting *mean_time*
9	*mean_az*	float(53)	f8.3	140-147	Azimuth mean
10	*mean_slow*	float(53)	f8.3	149-156	Slowness mean
11	*mean_time*	float(53)	f8.3	158-165	Arrival time mean
12	*sd_az*	float(53)	f6.1	167-172	Normalized azimuth residual
13	*sd_slo*	float(53)	f6.3	174-179	Normalized slowness residual
14	*sd_time*	float(53)	f6.3	181-186	Normalized arrival time residual
15	*lddate*	date	a19	188-206	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta*
Data:	Measurement	*detections, assoc_dets, added_dets, dets_az, dets_slo, dets_time, mean_az, mean_slo, mean_time, sd_az, sd_slo, sd_time*
	Administrative	*lddate*

station_type

The **station_type** table contains performance monitoring station type information.

Table A68. station_type

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station
2	*statype*	varchar2(4)	a4	8-11	Station type
3	*ncomp*	number(8)	i8	13-20	Number of components
4	*lddate*	date	a19	22-40	Load date

Category:	Performance Monitoring	
CSCI(s)	Performance Monitoring, Tuning Tools	
Keys:	Primary	*sta*
Data:	Descriptive	*statype, ncomp*
	Administrative	*lddate*

timestamp

The **timestamp** table is used by automated processing to record time milestones associated with time-series data.

Table A69. timestamp

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*procclass*	varchar2(16)	a16	1-16	Process class
2	*procname*	varchar2(16)	a16	18-33	Process name
3	*time*	float(53)	f17.5	35-51	Last epoch time
4	*lddate*	date	a19	53-71	Load date

Category:	Distributed Processing	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Tuning Tools	
Keys:	Primary	*procclass/procname*
Data:	Descriptive	*procclass, procname, time*
	Administrative	*lddate*

wfactivity

The **wfactivity** table describes activity in the **wfdisc** table for a channel group and time region.

Table A70. wfactivity

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*class*	varchar2(16)	a16	1-16	Type of interval
2	*name*	varchar2(20)	a20	18-37	Name of interval
3	*time*	float(53)	f17.5	39-55	Last epoch time
4	*duration*	float(24)	f7.2	57-63	Duration in seconds of the time region
5	*min_time*	float(53)	f17.5	65-81	Minimum time found in **wfdisc** for the time period
6	*max_endtime*	float(53)	f17.5	83-99	Maximum *endtime* found in **wfdisc** for the time period
7	*moddate*	date	a17	101-117	Time of last processing state change
8	*lddate*	date	a19	119-137	Load date

Category:	Data Archiving	
CSCI(s)	Data Management	
Keys:	Primary	*class/name/time*
Data:	Descriptive	*duration, min_time, max_endtime*
	Administrative	*moddate, lddate*

wfaudit

The **wfaudit** table contains records describing the sequences of changes made to rows in the **wfdisc** table for continuous (raw) waveform data.

Table A71. wfaudit

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*slotid*	number(8)	i8	1-8	Slot identifier
2	*chanid*	number(8)	i8	10-17	Channel identifier
3	*old_time*	float(53)	f17.5	19-35	*:old.time* value provided to the **wfdisc** trigger
4	*new_time*	float(53)	f17.5	37-53	*:new.time* value provided to the **wfdisc** trigger
5	*old_endtime*	float(53)	f17.5	55-71	*:old.endtime* provided to the **wfdisc** trigger
6	*new_endtime*	float(53)	f17.5	73-89	*:new.endtime* provided to the **wfdisc** trigger
7	*moddate*	date	a17	91-107	Time of last processing state change

Category:	Data Archiving	
CSCI(s)	Data Management	
Keys:	Primary	*slotid*
	Foreign	*chanid*
Data:	Descriptive	*old_time, new_time, old_endtime, new_endtime*
	Administrative	*moddate*

wfconv

The **wfconv** table contains data translations that are to be performed on incoming data before they are written to disk by the *DLMan* application. Data compression types include "-" if the data is not compressed or "CA" for Canadian compression. "Type" in columns *intype* and *outtype* is the fixed-width data type (for example, "s4") or "-" if not applicable (that is, if the data is compressed).

An *insamp* value of zero (0) indicates that the number of samples varies. Values less than zero in columns *insamp* and *outsamp* indicate that the total number of samples must be evenly divisible by *insamp*. *Strip* tells whether to strip the authentication headers from the data; "y" means strip them and "n" means do not strip them.

Table A72. wfconv

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*chanid*	number(8)	i8	17-24	Channel identifier
4	*inauth*	varchar2(1)	a1	26-26	Input authenticated (y or n)
5	*incomp*	varchar2(2)	a2	28-29	Input compression type
6	*intype*	varchar2(2)	a2	31-32	Input fixed-width datatype
7	*insamp*	number(8)	i8	34-41	Input samples per packet
8	*outauth*	varchar2(1)	a1	43-43	Output authenticated (y or n)
9	*outcomp*	varchar2(2)	a2	45-46	Output compression type
10	*outtype*	varchar2(2)	a2	48-49	Output fixed-width datatype
11	*outsamp*	number(8)	i8	51-58	Output samples per packet

Table A72. wfconv (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
12	*strip*	varchar2(1)	a1	60-60	Data stripped of headers
13	*commid*	number(8)	i8	62-69	Comment identifier
14	*lddate*	date	a19	71-89	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*sta/chan*
	Alternate	*chanid*
	Foreign	*commid*
Data:	Descriptive	*sta, chan, inauth, incomp, intype, insamp, outauth, outcomp, outtype, outsamp, strip*
	Administrative	*lddate*

wfdisc

The **wfdisc** table contains a waveform header file and descriptive information. This table provides a pointer (or index) to waveforms stored on disk. The waveforms themselves are stored in ordinary disk files called **wfdisc** or ".w" files as a sequence of sample values (usually in binary representation).

Table A73. wfdisc

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*chan*	varchar2(8)	a8	8-15	Channel code
3	*time*	float(53)	f17.5	17-33	Epoch time of first sample in file
4	*wfid*	number(9)	i9	35-43	Waveform identifier
5	*chanid*	number(8)	i8	45-52	Channel identifier
6	*jdate*	number(8)	i8	54-61	Julian date
7	*endtime*	float(53)	f17.5	63-79	Time + (nsamp-1)/ samprate
8	*nsamp*	number(8)	i8	81-88	Number of samples
9	*samprate*	float(24)	f11.7	90-100	Sampling rate in samples/sec
10	*calib*	float(24)	f16.6	102-117	Nominal calibration
11	*calper*	float(24)	f16.6	119-134	Nominal calibration period
12	*instype*	varchar2(6)	a6	136-141	Instrument code
13	*segtype*	varchar2(1)	a1	143-143	Indexing method
14	*datatype*	varchar2(2)	a2	145-146	Numeric storage
15	*clip*	varchar2(1)	a1	148-148	Clipped flag
16	*dir*	varchar2(64)	a64	150-213	Directory
17	*dfile*	varchar2(32)	a32	215-246	Data file

Table A73. wfdisc (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
18	*foff*	number(10)	i10	248-257	Byte offset of data segment within file
19	*commid*	number(9)	i9	259-267	Comment identifier
20	*lddate*	date	a19	269-287	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Services, Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*sta/chan/time*
	Alternate	*wfid*
	Foreign	*chanid, commid*
Data:	Descriptive	*sta, chan, dir, dfile, foff*
	Measurement	*time, jdate, endtime, nsamp, samprate, calib, calper, instype, segtype, datatype, clip*
	Administrative	*lddate*

wftag

The **wftag** table links various identifiers (for example, *orid, arid,* and *stassid* to *wfid*). Linkages can also be determined indirectly using *sta/chan/time*; however, it is more efficient to use the **wftag** table.

Table A74. wftag

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*tagname*	varchar2(8)	a8	1-8	Key (*arid*, *orid*, *evid*, etc.)
2	*tagid*	number(9)	i9	10-18	Tagname value
3	*wfid*	number(9)	i9	20-28	Waveform identifier
4	*lddate*	date	a19	30-48	Load date

Category:	Fundamental	
CSCI(s)	Data Management, Distributed Application Control System, Automatic Processing, Interactive Processing, Tuning Tools	
Keys:	Primary	*tagname/tagid/wfid*
Data:	Descriptive	*tagname*
	Administrative	*lddate*

wftape

The **wftag** table performs a similar function as **wfdisc**. Rather than pointing to the location of .w or waveform files stored on the disk, it points to waveforms stored on tape.

Table A75. wftape

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*sta*	varchar2(6)	a6	1-6	Station code
2	*chan*	varchar2(8)	a8	8015	Channel code
3	*time*	float(53)	f17.5	17-33	Epoch time of first sample in file
4	*wfid*	number(9)	i9	35-43	Waveform identier
5	*chanid*	number(8)	i8	45-52	Channel operation identifier
6	*jdate*	number(8)	i8	54-61	Julian date
7	*endtime*	float(53)	f17.5	63-79	Time + (nsamp-1)/ samprate
8	*nsamp*	number(8)	i8	81-88	Number of samples
9	*samprate*	float(24)	f11.7	90-100	Sampling rate in samples/sec
10	*calib*	float(24)	f16.6	102-117	Nominal calibration
11	*calper*	float(24)	f16.6	119-134	Nominal calibration period
12	*instype*	varchar2(6)	a6	136-141	Instrument code
13	*segtype*	varchar2(1)	a1	143-143	Indexing method
14	*datatype*	varchar2(2)	a2	145-146	Numeric storage
15	*clip*	varchar2(1)	a1	148-148	Clipped flag
16	*dir*	varchar2(64)	a64	150-213	Directory
17	*dfile*	varchar2(32)	a32	215-246	Data file
18	*foff*	number(9)	i10	248-257	Byte offset of data segment within file
19	*commid*	number(9)	i9	259-267	Comment identifier

Table A75. wftape (Continued)

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
20	*tapename*	varchar2(32)	a32	269-300	Tape name
21	*fileno*	number(4)	i4	302-305	Tape file number
22	*lddate*	date	a19	307-325	Load date

Category:	Continuous Data Subsystem	
CSCI(s)	Data Management	
Keys:	Primary	*sta/chan/time*
	Alternate	*wfid*
	Foreign	*chanid, commid*
Data:	Descriptive	*sta, chan, dir, dfile, foff, tapename, fileno*
	Measurement	*time, jdate, endtime, nsamp, samprate, calib, calper, instype, segtype, datatype, clip*
	Administrative	*lddate*

xtag

The **xtag** table links various identifiers (for example, *orid, arid, stassid,* and *wfid*) to other identifiers. This table is a generalization of the **wftag** table, which is limited to linking exclusively to the **wfid** table. The *thisdb* column describes the database account for the record specified by *thisid* and *thisname*; *thatdb* describes the database account for the record specified by *thatid* and *thatname*. When a parent/child table exists between the records *thisid* should designate the parent and *thatid* should designate the child.

Table A76. xtag

FIELD NUMBER	COLUMN	STORAGE TYPE	EXTERNAL FORMAT	CHARACTER POSITION	DESCRIPTION
1	*thisid*	number(9)	i9	1-9	*thisname* identifier
2	*thatid*	number(9)	i9	11-19	*thatname* identifier
3	*thisname*	varchar2(8)	a8	21-28	Key for *thisid* (*grid, orid, ntid*, and so on)
4	*thatname*	varchar2(8)	a8	30-37	Key for *thatid* (*arid, orid, nfid*, and so on)
5	*thisdb*	varchar2(32)	a32	39-70	Database account for the records specified by *thisid* and *thisname*
6	*thatdb*	varchar2(32)	a32	72-103	Database account for the records specified by *thatid* and *thatname*
7	*lddate*	date	a19	105-123	Load date

Category:	Message Subsytem	
CSCI(s)	Data Services, Data Management	
Keys:	Primary	*thisid/thisname*
Data:	Descriptive	*thisid, thatid, thisname, thatname, thisdb, thatdb*
	Administrative	*lddate*

Appendix B. Column Descriptions

This page is included in this document's electronic file as a placeholder for development of Table of Contents purposes only. The electronic version of this appendix is a separate file.

Appendix B. Column Descriptions

This appendix describes the columns in the tables used in the US NDC database schema and includes the following topics:

- Ranges
- NA Values
- Conventions
- Column Definitions

B.1 Ranges

Wherever possible, an explicit range is defined for each column. This range is important for data integrity and database management systems that automatically check ranges. When the range consists of a relatively small number of discreet values, the following notation is used:

column ∈ {`value-1`, `value-2`, …, `value-n`}

No range is documented for columns whose value may be any character string.

B.2 NA Values

Sometimes no information is available for a column. In that case, a Not Available (NA) Value is assigned. An NA Value is outside the range of permissible or recommended values for the column. This special NA Value alerts users and applications that the desired column was not available when the record was created. For example, in the **origin** table, the column *ms* (surface wave magnitude) may be unknown for a given row. Then the NA Value for magnitudes (–999.0) should be assigned to *ms* and *msid* should be set to –1, the NA Value for *msid*. Some columns are essential to defining a meaningful record, and they must be specified; the NA Value is not permitted. For example, the column time in **arrival** must be given a value in the valid range, not an NA Value. Another example is magnitude in **stamag**. Magnitude must be given a meaningful value for each record so no NA Value is defined.

Some general guidelines and specific examples of NA Values are given in Table B1. These are only guidelines and NA Values may not be unique to a particular column.

Table B1. Guidelines and Examples of NA Values

COLUMN TYPE/RANGE:	NA VALUE:	EXAMPLES
Character columns	- (hyphen)	*bmtyp*, *auth*
Non-negative integers	`-1`	*chanid*, *arid*
Non-negative real numbers	`-1.0`	*cfreq*, *deltim*
Real numbers > `-999.0`	`-999.0`	*azres*
Large real numbers	`-9999999999.999` or `+9999999999.999`	*endtime*, *time*

An NA Value should not be confused with an ORACLE NULL. NA Values are supplied by users, while ORACLE inserts the database value NULL when no value is specified. An column containing a database value of NULL appears blank when selected within SQL*Plus. When creating a table, an column may be constrained as NOT NULL to require the user to supply a value. The ORACLE `describe` command will identify such columns as NOT NULL. No correlation is intended between ORACLE NOT NULL requirements and the US NDC requirements that an column must be specified.

B.3 Conventions

This section uses graphical and typographical conventions as described in Table B2.

Table B2. Typographical Conventions

ELEMENT	APPEARANCE	EXAMPLE
Database table Database table and columns, when written in the dot notation	**Bold**	**dataready** **prodtrack.status**
Database columns Processes, software units, and libraries Titles of documents	Italics	*status* *ARS, libpar* *GA Subsystem Software*
Value of a key or component of a key	Courier font	`Orid`
Database accounts and database names	All capital letters	GLOBAL OPSDB
SQL*Plus statements or commands	Underline	Select

B.4 Column Definitions

The definitions for the US NDC databases follow.

Name:	*added*	
Table:	**ex_an**	
Description:	Number of phases added by an analyst to an expert system event solution. An added phase is an arrival not available to the expert system.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*added* ≥ 0	
Name:	*added_dets*	
Table:	**station_hist**	
Description:	Number of added detections normalized for one day	
Format:	float(24)	External: f10.2
NA Value:	NOT ALLOWED	
Range:	*added_dets* > 0.0	
Name:	*address*	
Table:	**alphasite**	
Description:	Internet protocol (IP) address of source of continuous data	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	*address* ∈ {0.0.0.0 – 255.255.255.255}	
Name:	*adef*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of associated nondefining phases. The observations for these phases are not used in the location solution.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*adef* ≥ 0	
Name:	*algorithm*	
Table:	**origerr (origerr_ga)**	
Description:	Location algorithm used. This column is a brief textual Description: of the algorithm used for computing a seismic origin.	
Format:	varchar2(15)	External: a15
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	

Name:	*amcor*		
Table:	**siteaux**		
Description:	Site-dependent log amplitude correction		
Format:	float(24)	External:	f10.1
NA Value:	`-999.0`		
Range:	*amcor* > –999.0		
Name:	*amcorsd*		
Table:	**siteaux**		
Description:	Standard deviation for log amplitude correction		
Format:	float(24)	External:	f5.2
NA Value:	`-1.0`		
Range:	*amcorsd* > 0.0		
Name:	*amp*		
Table:	**amplitude**, **arrival**		
Description:	Measured amplitude defined by *amptype*		
Format:	float(24)	External:	f11.2
NA Value:	`-1.0`		
Units:	Nanometers or dimensionless depending on the type of channel		
Range:	*amp* > 0.0		
Name:	*ampcorclip*		
Table:	**hydro_arrival**		
Description:	Correction to raw amplitude for clipping		
Format:	float(24)	External:	f11.4
NA Value:	`-1.0`		
Units:	Amplitude		
Range:	*ampcorclip* > 0.0		
Name:	*ampcordepth*		
Table:	**hydro_arrival**		
Description:	Correction to raw amplitude for depth		
Format:	float(24)	External:	f11.4
NA Value:	`-1.0`		
Units:	Kilometers		
Range:	*ampcordepth* > 0.0		
Name:	*ampcordist*		
Table:	**hydro_arrival**		
Description:	Correction to raw amplitude for distance		
Format:	float(24)	External:	f11.4
NA Value:	`-1.0`		
Units:	Degrees		
Range:	*ampcordist* > 0.0		

Name:	*ampdescr*	
Table:	**ampdescript**	
Description:	Amplitude measurement parameters descriptions	
Format:	varchar2(255)	External: a255
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*ampid*	
Table:	**amplitude**, **stamag**	
Description:	Amplitude identifier. Every amplitude measure is assigned a unique positive integer that identifies it in the database. If an associated **stamag** record exists, then *ampid* links it to **amplitude**.	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED `-1` in **stamag**	
Range:	*ampid* > 0	

Name:	*amplr*	
Table:	**apma**	
Description:	Maximum 3-component amplitude for all overlapping time windows used in the polarization analysis. This column is equal to the sum of the square roots of the eigenvalues. The only difference between *amps* and *amplr* is in the definition of the overlapping time windows.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Nanometers	
Range:	*amplr* > 0.0	

Name:	*ampp*	
Table:	**apma**	
Description:	3-component amplitude measured at the time of the maximum rectilinearity. This column is equal to the sum of the square roots of the eigenvalues (that is, it is the sum of the amplitudes measured along the three axes of the polarization ellipsoid).	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Nanometers	
Range:	*ampp* > 0.0	

Name:	*amps*	
Table:	**apma**	
Description:	Maximum 3-component amplitude for all overlapping time windows used in the polarization analysis. This column is equal to the sum of the square roots of the eigenvalues. The only difference between *amps* and *amplr* is in the definition of the overlapping time windows.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Nanometers	
Range:	*amps* > 0.0	

Name:	*amptime*	
Table:	**amplitude**	
Description:	Epoch time of amplitude measure	
Format:	float(53)	External: f17.5
NA Value:	`-9999999999.999`	
Units:	Seconds	
Range:	*amptime* > –9999999999.999	
Name:	*amptype*	
Table:	**ampdescript**, **amplitude**	
Description:	Amplitude measure descriptor. This descriptor is used to uniquely identify an amplitude measurement and link the description in **ampdescript** with actual measurements in **amplitude**.	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*aoi_geochar*	
Table:	**aoi**	
Description:	Area of interest geographic region characteristic. This column describes the geographic region in which an event is located. This type specifies whether the event is located in an area of interest (`i`) or outside the area of interest (`o`) (see *depth_geochar*, *seismic_geochar*, and *terrain_geochar*).	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*aoi_geochar* ∈ {`i` | `o`}	
Name:	*apmarid*	
Table:	**apma**	
Description:	Unique **apma** recipe identifier. Each arrival in **apma** is assigned a positive integer identifying it with the recipe used in the polarization analysis.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*apmarid* > 0	
Name:	*archived*	
Table:	**dlfile**	
Description:	Status of data archiving: Archiving (`a`), yes (`y`), or no (`n`)	
Format:	varchar2(1)	External: a1
NA Value:	– (hyphen)	
Range:	*archived* ∈ {`a`, `n`, `y`}	
Name:	*archiveport*	
Table:	**dlman**	
Description:	Archiver port	
Format:	number(6)	External: i6
NA Value:	`-1`	
Range:	`0` ≤ *archiveport* ≤ `16383`	

Name:	*arid*	
Table:	**amp3c**, **amplitude**, **apma, arrival**, **assoc (assoc_ga), detection**, **hydro_arrival**, **hydro_assoc**, **stamag**	
Description:	Arrival identifier. Each arrival is assigned a unique positive integer identifying it with a unique *sta*, *chan*, and *time*.	
Format:	number(p)	External: i9
NA Value:	NOT ALLOWED `-1` for **stamag**	
Range:	*arid* > 0	
Name:	*array*	
Table:	**request**	
Description:	Array code. The network or station name	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size matching the net column in **affiliation (stanet)**	
Name:	*assoc*	
Table:	**missed_class**	
Description:	Number of associated detections in the second bulletin not detected in the first bulletin	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*assoc* ≥ 0	
Name:	*assoc_dets*	
Table:	**station_hist**	
Description:	Number of associated detections normalized for one day	
Format:	float(24)	External: f11.2
NA Value:	`-1.0`	
Units:	Nanometers or dimensionless depending on the type of channel	
Range:	*amp* > 0.0	
Name:	*asstr*	
Table:	**bull_comp**	
Description:	Association strength of two events: strong (`s`) or weak (`w`). An origin (**origin1**) is strongly associated with an origin in the other database account (**origin2**) if three or more defining detections for **origin1** are also associated with **origin2**, or all defining detections for **origin1** are also associated with **origin2**. If events are associated only by time and location (no arrivals available) then *asstr* is set to `w`.	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*asstr* ∈ {`s`, `w`}	

Name:	*asta*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of associated arrivals from regional arrays. Regional is currently defined as a station-event distance not less than 250 km and up to 2,000 km.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*asta* ≥ 0	
Name:	*auth*	
Table:	**amplitude, apma, arrival, event, interval, netmag, network, origin (origin_ga), pixdisc, qcdata, siteaux, stamag**	
Description:	Author, the originator of the data; may also identify an application generating the record, such as an automated interpretation or signal-processing program.	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*avgconstval*	
Table:	**qcstats**	
Description:	Average number of seconds in masked constant segments	
Format:	float(53)	External: f17.5
NA Value:	`-999.0`	
Units:	Same as waveform data	
Range:	*aveconstval* ≥ `0.0`	
Name:	*az1*	
Table:	**hydro_arr_group**	
Description:	Azimuth estimated from the time lags of arrivals in a hydro-arrival group.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *az1* < 360.0	
Name:	*az2*	
Table:	**hydro_arr_group**	
Description:	Azimuth estimated from the time lags of arrivals in a hydro-arrival group. This second azimuth estimate is only needed when only two arrivals exist in a group, which results in an ambiguity between two equally likely azimuths. The error is the same for the two azimuths.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *az2* < 360.0	

Name:	*azcontrib*	
Table:	**hydro_assoc**	
Description:	Flag specifies if an arrival that belongs to a hydro-arrival group was used to calculate the azimuth	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*azcontrib* ∈ {y, n}	
Name:	*azdef*	
Table:	**assoc (assoc_ga)**	
Description:	Azimuth-defining code; one-character flag indicates whether or not the azimuth of a phase was used to constrain the event location solution. This column is defining (*azdef* = d) if it was used in the location, nondefining (*azdef* = n) if it was not.	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*azdef* ∈ {d, n}	
Name:	*azimuth*	
Table:	**arrival**	
Description:	Observed azimuth. This value is the estimated station-to-event azimuth measured clockwise from North. The estimate is made from f-k or polarization analysis.	
Format:	float(24)	External: f7.2
NA Value:	-1.0	
Units:	Degrees	
Range:	0.0 ≤ *azimuth* < 360.0	
Name:	*azres*	
Table:	**assoc (assoc_ga)**	
Description:	Azimuth residual. This value is the difference between the measured station-to-event azimuth for an arrival and the true azimuth. The true azimuth is the bearing to the inferred event origin.	
Format:	float(24)	External: f7.1
NA Value:	-999.0	
Units:	Degrees	
Range:	-180.0 ≤ *azres* ≤ 180.0	

Name:	*band*	
Table:	**instrument**	
Description:	Frequency band. This value is a qualitative indicator of frequency passband for an instrument. Values should reflect the response curve rather than just the sample rate. Recommended values are as follows: `s` (short-period) `m` (mid-period) `i` (intermediate-period) `l` (long-period) `b` (broadband) `h` (high-frequency, very short-period) `v` (very long-period) For a better notion of the instrument characteristics, see the instrument response curve.	
Format:	varchar2(1)	External: a1
NA Value:	`-` (hyphen)	
Range:	*band* ∈ {`s`, `m`, `i`, `l`, `b`, `h`, `v`}	
Name:	*bandw*	
Table:	**detection**	
Description:	Frequency bandwidth	
Format:	float(24)	External: f7.3
NA Value:	`-1.0`	
Units:	Hertz (Hz)	
Range:	*bandw* > 0.0	
Name:	*belief*	
Table:	**assoc (assoc_ga)**	
Description:	Phase identification confidence level. This value is a qualitative estimate of the confidence that a seismic phase is correctly identified.	
Format:	float(24)	External: f4.2
NA Value:	`-1.0`	
Range:	`0.0` ≤ *belief* ≤ `1.0`	
Name:	*bmtyp*	
Table	**detection**	
Description:	String indicating a coherent (`coh`), incoherent (`inc`), or horizontal (`hor`) beam type	
Format:	varchar2(4)	External: a4
NA Value:	`-` (hyphen)	
Range:	*bmtyp* ∈ {`coh`, `inc`, `hor`}	

Name:	*bordercolor*	
Table:	**mapdisc**	
Description:	Map border color name. A solid colored border may appear on the top, bottom, and right of any raster map.	
Format:	varchar2(32)	External: a32
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that forms a valid X11 color name (for example, `black`)	
Name:	*bpfrqac*	
Table:	**hydro_arrival**	
Description:	Auto correlation bubble pulse	
Format:	float(24)	External: f11.4
NA Value:	`-1.0`	
Range:	*bpfrqac* ≥ `0.0`	
Name:	*bpfrqcep*	
Table:	**hydro_arrival**	
Description:	Cepstrum bubble pulse	
Format:	float(24)	External: f11.4
NA Value:	`-1.0`	
Range:	*bpfrqcep* ≥ `0.0`	
Name:	*bullcomp*	
Table:	**datacollected**	
Description:	*BullComp* Description:	
Format:	varchar2(30)	External: a30
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*bulletins*	
Table:	**bull_comp, missed_class**	
Description:	Bulletins compared	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*calib*	
Table:	**calibrate, wfdisc (wfproto)**	
Description:	Calibration factor. This value is the conversion factor that maps digital data to earth displacement. The factor holds true at the oscillation period specified by the column *calper*. A positive value means ground motion increasing in component direction (up, North, East) is indicated by increasing counts. A negative value means the opposite. The column *calib* generally reflects the best calibration information available at the time of recording, but refinement may be given in **sensor**, reflecting a subsequent recalibration of the instrument (see *calratio*).	
Format:	float(24)	External: f16.6
NA Value:	NOT ALLOWED	
Units:	Nanometers/digital count	
Range:	*calib* > 0.0	
Name:	*calper*	
Table:	**calibrate, sensor, wfdisc (wfproto)**	
Description:	Calibration period; gives the period for which *calib*, *ncalib*, and *calratio* are valid.	
Format:	float(24)	External: f16.6
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*calper* > 0.0	
Name:	*calratio*	
Table:	**sensor**	
Description:	Calibration conversion ratio. The value is a dimensionless calibration correction factor that permits small refinements to the calibration correction made using *calib* and *calper* from the **wfdisc (wfproto)** table. Often, the **wfdisc (wfproto)** *calib* contains the nominal calibration assumed at the time of data recording. If the instrument is recalibrated, *calratio* provides a mechanism to update calibrations from **wfdisc (wfproto)** with the new information without modifying the **wfdisc (wfproto)** table. A positive value means ground motion increasing in component direction (up, North, East) is indicated by increasing counts. A negative value means the opposite. The column *calratio* is meant to reflect the most accurate calibration information for the time period for which the sensor record is appropriate, but the nominal value may appear until other information is available.	
Format:	float(24)	External: f16.6
NA Value:	NOT ALLOWED	
Range:	*calratio* > 0.0	
Name:	*cfreq*	
Table:	**amp3c, detection**	
Description:	Center frequency of a beam or f-k spectrum	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Hertz	
Range:	*cfreq* > 0.0	

Name:	*chan*	
Table:	**amplitude, arrival, calibrate, detection, dlfile, qcstats, request, sensor, siteaux, sitechan, wfconv, wfdisc (wfproto)**	
Description:	Channel code; an eight-character code which, taken together with *sta*, *jdate* and *time*, uniquely identifies seismic time-series data, including the geographic location, spatial orientation, sensor, and subsequent data processing (beam channel descriptor)	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED except in **arrival** where - (hyphen) is allowed	
Range:	Any character string up to the column size	
Name:	*chanid*	
Table:	**arrival**, **calibrate**, **dlfile**, **sensor**, **sitechan**, **wfaudit**, **wfconv**, **wfdisc (wfproto)**	
Description:	Channel identifier. This value is a surrogate key used to uniquely identify a specific recording. The column *chanid* duplicates the information of the compound key *sta*/*chan*/*time*.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*chanid* > 0	
Name:	*class*	
Table:	**chan_groups, interval, request, wfactivity**	
Description:	Request type for **request** table or interval type for **chan_groups**, **interval**, and **wfactivity** tables	
Format:	varchar2(16)	External: a16
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*clip*	
Table:	**amplitude**, **arrival**, **wfdisc (wfproto)**	
Description:	Clipped data flag. This value is a single-character flag to indicate whether (c) or not (n) the data was clipped	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*clip* ∈ {c, n}	
Name:	*colormapid*	
Table:	**colordisc**, **mapcolor**	
Description:	Colordisc identifier. Each colordisc is assigned a unique positive integer that identifies it in a database. The column *colormapid* identifies color-lookup tables available to maps.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*colormapid* > 0	

Name:	*colormapname*	
Table:	**colordisc**	
Description:	Colormap name that identifies the color-lookup table in a listing of available tables.	
Format:	varchar2(64)	External: a64
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*colorname*	
Table:	**overlaydisc**	
Description:	Overlay color name	
Format:	varchar2(32)	External: a32
NA Value:	– (hyphen)	
Range:	Any character string up to the column size that is a valid color name	
Name:	*command*	
Table:	**msgaux**	
Description:	Command that was being executed when the failure occurred. If this cannot be determined, such as a caught signal from UNIX, then the value is set to signal caught.	
Format:	varchar2(24)	External: a24
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	
Name:	*commid*	
Table:	**alphasite**, **apma**, **arrival**, **assoc** (**assoc_ga**), **channame**, **datauser**, **detection**, **discrimuse**, **discrimvote**, **dlfile**, **dlman**, **event**, **hydro_arrival**, **hydro_origin**, **msgdisc**, **netmag**, **network**, **origerr** (**origerr_ga**), **origin** (**origin_ga**), **qcdata**, **remark**, **siteaux**, **stamag**, **wfconv**, **wfdisc (wfproto)**	
Description:	Comment identifier. This value is a key that points to free-form comments entered in the **remark** table. These comments store additional information about a record in another table. The **remark** table can have many records with the same *commid* and different *lineno*, but the same *commid* will appear in only one other record among the rest of the tables in the database (see *lineno*).	
Format:	number(9)	External: i9
NA Value:	–1 NOT ALLOWED for **remark**	
Range:	*commid* > 0	
Name:	*complete*	
Table:	**request**	
Description:	Percentage complete. The percentage of waveform data acquired for this request	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	0 ≤ *complete* ≤ 100	

Name:	*conf*	
Table:	**origerr**	
Description:	Confidence measure for a particular event identification method	
Format:	float(24)	External: f5.3
NA Value:	NOT ALLOWED	
Range:	$0.5 \leq conf \leq 1.0$	

Name:	*connmanport*	
Table:	**dlman**	
Description:	Connection Manager (ConnMan) port number used to send messages to the diskloop manager application	
Format:	number(6)	External: i6
NA Value:	-1	
Range:	$0 \leq connmanport \leq 16383$	

Name:	*const*	
Table:	**qcstats**	
Description:	Amount of data in detection processing interval masked due to constant segments	
Format:	float(53)	External: f17.5
NA Value:	-999.0	
Units:	Seconds	
Range:	$const \geq 0.0$	

Name:	*constrain_depth*	
Table:	**event_control**	
Description:	Logical descriptor that tells location process whether or not to fix (constrain) the current hypocentral depth. If TRUE (1), the depth will be fixed to the value specified on the first (summary) line of the DATA file or as specified by the depth column of the **origin (origin_ga)** table. If FALSE (0), the depth is an independent solution parameter. Default is TRUE (1).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	$constrain\_depth \in \{0, 1\}$	

Name:	*constrain_latlon*	
Table:	**event_control**	
Description:	Logical descriptor that tells location process whether or not to fix (constrain) the current epicentral location. If TRUE (1), the latitude and longitude will be fixed to the value specified by the *lat* and *lon* columns of the **origin (origin_ga)** table. If FALSE (0), the latitude and longitude are independent solution parameters. Default is FALSE (0).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	$constrain\_latlon \in \{0, 1\}$	

Name:	*constrain_ot*	
Table:	**event_control**	
Description:	Logical descriptor that tells location process whether or not to fix (constrain) the current origin time. If TRUE (1), the origin time will be fixed to the value specified by the *time* column of the **origin** table. If FALSE (0), the origin time is an independent solution parameter. Default is FALSE (0).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	*constrain_ot* ∈ {0, 1}	

Name:	*controlport*	
Table:	**dlman**	
Description:	DataControl port. The number used to send commands to the diskloop manager application	
Format:	number(6)	External: i6
NA Value:	-1	
Range:	0 ≤ *controlport* ≤ 16383	

Name:	*cov_depth_time*	
Table:	**event_control**	
Description:	Coverage ellipse depth/time conversion factor. This value is the conversion factor to be multiplied by the depth and origin time axes (*sdepth* and *stime*) of the confidence ellipse to recover the coverage ellipse without having to do a complete relocation.	
Format:	float(24)	External: f9.4
NA Value:	-999.0	
Range:	*cov_depth_time* > 0.0	

Name:	*cov_sm_axes*	
Table:	**event_control**	
Description:	Coverage ellipse semi-axis conversion factor. This value is the conversion factor to be multiplied by the semi-major and semi-minor axes of the confidence ellipse to recover the coverage ellipse without having to do a complete relocation.	
Format:	float(24)	External: f9.4
NA Value:	-999.0	
Range:	*cov_sm_axes* > 0.0	

Name:	*cplag*	
Table:	**hydro_arrival**	
Description:	Crossing-point lag of the signal. The first zero crossing after the initial peak in the auto correlation of the signal	
Format:	float(24)	External: f11.4
NA Value:	-1.0	
Units:	Seconds	
Range:	*cplag* > 0.0	

Name:	*ctype*	
Table:	**sitechan**	
Description:	Channel type. This column specifies the type of data channel: normal (n) -- a normal instrument response, beam (b) -- a coherent beam formed with array data, or incoherent (i) -- an incoherent beam or energy stack.	
Format:	varchar2(4)	External: a4
NA Value:	- (hyphen)	
Range:	*ctype* ∈ {n, b, i}	
Name:	*dasta*	
Table:	**ex_an**	
Description:	Difference in number of regional array stations contributing to the analyst and expert system origins. This value is [*asta* (analyst) – *asta* (expert system)] for analyst versus expert system comparisons or [*asta* (bulletin1) – *asta* (bulletin2)] for more general bulletin comparisons	
Format:	number(8)	External: i8
NA Value:	-999	
Range:	*dasta* > –999	
Name:	*database1*	
Table:	**datacollected**	
Description:	Name of first database used as the source of input information for a bulletin comparison	
Format:	varchar2(15)	External: a15
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid database name	
Name:	*database2*	
Table:	**datacollected**	
Description:	Name of second database used as the source of input information for a bulletin comparison	
Format:	varchar2(15)	External: a15
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid database name	

Name:	*datatype*
Table:	**wfdisc (wfproto)**
Description:	Numeric data storage. This column specifies the format of a data series in the file system. Data types: `i4`, `f4`, and `s4` are typical values. Datatype `i4` denotes a 4-byte integer and `f4` denotes a 32-bit real number. Datatype `s4` is an integer where the most significant byte is in the low address position in memory and is opposite to the `i4` order. Machine-dependent formats are supported for common hardware to allow data transfer in native machine binary formats. American Standard Code for Information Interchange (ASCII) formats have also been defined to retain full precision of any binary data type. ASCII may be used when exchanging data between computer systems with incompatible binary types (see the *wfport* command manual page for information about converting formats). Datatype can only describe single values or arrays of one data type.
Format:	varchar2(2) External: a2
NA Value:	- (hyphen)
Range:	Datatype ∈ `{a0,b0,c0,a#,b#,c#,t4,t8,s4,s2,s3,f4,f8,i4,i2,e#,g2}`

Value	Size (bytes)	Description:
a0	15	ASCII single precision
b0	24	ASCII double precision
c0	12	ASCII integer
a#	15	ASCII single precision
b#	24	ASCII double precision
c#	12	ASCII integer
t4	4	SUN Institute of Electrical and Electronics Engineers (IEEE) single precision real
t8	8	SUN IEEE double precision real
s4	4	SUN IEEE integer
s2	2	SUN IEEE short integer
s3	3	SUN IEEE integer
f4	4	VAX IEEE single precision real
f8	8	VAX IEEE double precision real
i4	4	VAX IEEE integer
i2	2	VAX IEEE short integer
e#	2048*#	Compressed data format
g2	2	Norwegian Regional Experimental Seismic System (NORESS) gain-ranged

Name:	*ddepth*
Table:	**bull_comp, ex_an**
Description:	Difference in depth between corresponding origin locations. For depth comparisons between analysts and the expert system, the value is [*depth* (analyst) – *depth* (expert system)]. For more general bulletin comparisons, the value is [*depth* (bulletin1) – *depth* (bulletin2)].
Format:	float(24) External: f6.1
NA Value:	`-999.0`
Units:	Kilometers
Range:	*ddepth* > –999.0

Name:	*ddepthp*
Table:	**ex_an**
Description:	Difference in the number of defining depth phases associated with analyst and expert system origins. A depth phase is a member of the set {sP, pP, sS}. The value is: [(number-analyst-phases) – (number-expert-phases)].
Format:	number(8) External: i8
NA Value:	-999
Range:	*ddepthp* > –999
Name:	*ddist*
Table:	**bull_comp, ex_an**
Description:	Difference in distance between corresponding origins in a bulletin comparison
Format:	float(24) External: f8.3
NA Value:	-1.0
Units:	Kilometers
Range:	*ddist* ≥ 0.0
Name:	*deast*
Table:	**site**
Description:	Distance East. This column gives the easting or the relative position of an array element East of the location of the array center specified by the value of *refsta* (see *dnorth*).
Format:	float(24) External: f9.4
NA Value:	0.0
Units:	Kilometers
Range:	-20,000.0 ≤ *deast* ≤ 20,000.0
Name:	*delaz*
Table:	**arrival, detection, hydro_arr_group**
Description:	Azimuth uncertainty. This column is an estimate of the standard deviation of the azimuth of a signal
Format:	float(24) External: f7.2
NA Value:	-1.0
Units:	Degrees
Range:	*delaz* > 0.0
Name:	*delslo*
Table:	**arrival, detection**
Description:	Slowness uncertainty. This column is an estimate of the standard deviation of the slowness of a signal
Format:	float(24) External: f7.2
NA Value:	-1.0
Units:	Seconds/kilometers for **detection** Seconds/degree for **arrival**
Range:	*delslo* > 0.0

Name:	*delta*	
Table:	**assoc (assoc_ga), stamag**	
Description:	Source-receiver distance. This column is the arc length, over the Earth's surface, of the path the seismic phase follows from source to receiver. The location of the origin is specified in the origin (**origin_ga**) record referenced by the column *orid*. The column *arid* points to the record in the **arrival** table that identifies the receiver. The value of the column can exceed 360 degrees. The geographic distance between source and receiver is delta modulo(180).	
Format:	float(24)	External: f8.3
NA Value:	-1.0	
Units:	Degrees	
Range:	*delta* ≥ 0.0	
Name:	*deltaf*	
Table:	**amplitude**	
Description:	Sample interval width	
Format:	float(24)	External: f7.3
NA Value:	-1.0	
Range:	*deltaf* > 0.0	
Name:	*deltim*	
Table:	**arrival, detection**	
Description:	Arrival time uncertainty. This column is an estimate of the standard deviation of an arrival time.	
Format:	float(24)	External: f6.3
NA Value:	-1.0	
Units:	Seconds	
Range:	*deltim* > 0.0	
Name:	*depdp*	
Table:	**origin (origin_ga)**	
Description:	Depth as estimated from depth phases. This value is a measure of event depth estimated from a depth phase or an average of several depth phases. Depth is measured positive in a downwards direction, starting from the Earth's surface (see *ndp*).	
Format:	float(24)	External: f9.4
NA Value:	-999.0	
Units:	Kilometers	
Range:	0.0 ≤ *depdp* < 1000.0	
Name:	*depth*	
Table:	**origin (origin_ga)**	
Description:	Source depth. This column gives the depth (positive down) of the event origin. Negative depth implies an atmospheric event.	
Format:	float(24)	External: f9.4
NA Value:	-999.0	
Units:	Kilometers	
Range:	-100.0 ≤ *depth* < 1000.0	

Name:	*depth_geochar*	
Table:	**aoi**	
Description:	Depth geographic region characteristic. There are four characteristics that describe the geographic region in which an event is located. This type specifies whether the event is located in an area in which earthquakes historically occur at shallow (<100 km) depths or deep (>100 km) depths (see *aoi_geochar*, *seismic_geochar*, and *terrain_geochar*).	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*depth_geochar* ∈ {s | d}	
Name:	*depthp*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of time-defining depth phases. A depth phase is a member of the set {sP, pP, sS}.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*depthp* ≥ 0	
Name:	*descrip*	
Table:	**sitechan**	
Description:	Text Description:	
Format:	varchar2(50)	External: a50
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*dest*	
Table:	**mig_rules**	
Description:	Destination database for migration	
Format:	varchar2(10)	External: a10
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid name of database server	
Name:	*dest_tbl*	
Table:	**mig_rules**	
Description:	Destination table for database migration	
Format:	varchar2(30)	External: a30
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid table name	
Name:	*detections*	
Table:	**station_hist**	
Description:	Number of arrivals normalized to one day	
Format:	float(24)	External: f10.2
NA Value:	NOT ALLOWED	
Range:	*detections* > 0.0	

Name:	*detendtime*	
Table:	**qcstats**	
Description:	End time of actual interval used for detection processing	
Format:	float(53)	External: f17.5
NA Value:	`-9999999999.999`	
Units:	Seconds	
Range:	*detendtime* > -9999999999.999	
Name:	*dets_az*	
Table:	**station_hist**	
Description:	Number of detections affecting *mean_az*	
Format:	float(24)	External: f10.2
NA Value:	NOT ALLOWED	
Range:	*det_az* > 0.0	
Name:	*dets_slo*	
Table:	**station_hist**	
Description:	Number of detections affecting *mean_slo*	
Format:	float(24)	External: f10.2
NA Value:	NOT ALLOWED	
Range:	*det_slo* > 0.0	
Name:	*dets_time*	
Table:	**station_hist**	
Description:	Number of detections affecting *mean_time*	
Format:	float(24)	External: f10.2
NA Value:	NOT ALLOWED	
Range:	*dets_time* > 0.0	
Name:	*dettime*	
Table:	**qcstats**	
Description:	Start time of actual interval used for detection processing	
Format:	float(53)	External: f17.5
NA Value:	`+9999999999.999`	
Units:	Seconds	
Range:	*dettime* < -9999999999.999	
Name:	*dfid*	
Table:	**dlfile**	
Description:	Diskloop file identifier	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*dfid* > 0	

Name:	*dfile*	
Table:	**colordisc, dlfile, instrument, interval_files, mapdisc, msgdisc, overlaydisc, pixdisc, wfdisc (wfproto)**	
Description:	Name of data file	
Format:	varchar2(32)	External: a32
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that conforms to UNIX filename syntax	
Name:	*did*	
Table:	**ex_an**	
Description:	Difference in event type between the analyst and expert system origins (see *etype*); *did* is `y` if the event types are the same or `n` if the event types are different.	
Format:	varchar2(4)	External: a4
NA Value:	- (hyphen)	
Range:	*did* ∈ {`y`, `n`}	
Name:	*digital*	
Table:	**instrument**	
Description:	Flag denoting whether this instrument record describes an analog (`a`) or digital (`d`) recording system	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*digital* ∈ {`d`, `a`}	
Name:	*dimx*	
Table:	**mapdisc**	
Description:	Width (or x-dimension) of the map in pixels	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Units:	pixels	
Range:	*dimx* > 0	
Name:	*dimy*	
Table:	**mapdisc**	
Description:	Height (or y-dimension) of the map in pixels	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Units:	pixels	
Range:	*dimy* > 0	

Name:	*dir*
Table:	**colordisc, dlfile, instrument, interval_files, mapdisc, msgdisc, overlaydisc, pixdisc, wfdisc (wfproto)**
Description:	Directory. This column is the directory part of a path name. Relative path names or dot (`.`), the notation for the current directory, may be used. Directory to find file (**msgdisc**).
Format:	varchar2(64) External: a64
NA Value:	NOT ALLOWED
Range:	Any character string up to the column size that conforms to UNIX directory name syntax
Name:	*discrim_flag*
Table:	**discrimuse**
Description:	Discriminant flag. This column indicates whether or not the measurement of this discriminant was used in the overall event classification for this origin and station. Each station associated to an origin can potentially contribute to each of six different discriminants (*discrimtypes*) used to determine the event's overall classification (see *class_code* and *sem_code* for more details).
Format:	varchar2(1) External: a1
NA Value:	- (hyphen)
Range:	`discrim_flag ∈ {T, F}`

Name:	*discrimtype*		
Table:	**discrimuse, discrimvote**		
Description:	Discriminant type; currently can be based on one of nineteen methods:		
	Value	Description	
	CF	Complexity factor	
	FM	First motion	
	LP	Body wave, surface wave magnitudes (Mb minus Ms)	
	PP	Depth from reflected phases (pP, sP)	
	RESULT	Indicates final results of *discrim*	
	SP	Depth from S minus P	
	TT	Depth from hypocenter calculations (travel time)	
	BUB	Bubble pulse	
	DEEP	Deep seismicity	
	DU	Duration	
	FR	Filter ratio	
	HF	Hydro high frequency	
	HT	Hydro T phase	
	LO	Location	
	MINE	Mining activity	
	NA	Normalized amplitude	
	RTRMS	Amplitude of bubble pulse/rms noise in correlation	
	SEIS	Seismic area	
	VOL	Volcanic area	
Format:	varchar2(10)		External: a10
NA Value:	- (hyphen)		
Range:	Any character string up to the column size		
Name:	*dlid*		
Table:	**alphasite, dlfile, dlman**		
Description:	Diskloop manager identification		
Format:	number(8)		External: i8
NA Value:	NOT ALLOWED		
Range:	*dlid* ≥ 0		
Name:	*dlsta*		
Table:	**ex_an**		
Description:	In a comparison of bulletins, *dlsta* is the difference in the number of local stations contributing to the same event from the corresponding bulletins. The value is [*lsta* (analyst) – *lsta* (expert system)] for analyst versus expert system comparisons and [*lsta* (bulletin1) – *lsta* (bulletin2)] for more general bulletin comparisons.		
Format:	number(8)		External: i8
NA Value:	-999		
Range:	*dlsta* > –999		

Name:	*dnarr*	
Table:	**bull_comp**	
Description:	Absolute difference in the number of associated arrivals between corresponding origins	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*dnarr* ≥ 0	
Name:	*dndef*	
Table:	**bull_comp, ex_an**	
Description:	Difference in the number of defining phases between corresponding origins. A phase is defining only if its time-component is defining. The value is [*ndef* (analyst) – *ndef* (expert system)] for analyst versus expert system comparisons and [*ndef* (bulletin1) – *ndef* (bulletin2)] for more general bulletin comparisons.	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*dndef* ≥ 0	
Name:	*dnorth*	
Table:	**site**	
Description:	Distance North. This column gives the northing or relative position of array element North of the array center specified by the value of *refsta* (see *deast*)	
Format:	float(24)	External: f9.4
NA Value:	0.0	
Units:	Kilometers	
Range:	-20,000.0 ≤ *dnorth* ≤ 20,000.0	
Name:	*dnsta*	
Table:	**ex_an**	
Description:	In a comparison of bulletins, *dnsta* is the difference in the number of contributing stations between the corresponding bulletin locations. The value is [*nsta* (analyst) – *nsta* (expert systems)] for analyst versus expert system comparisons and [*nsta* (bulletin1) – *nsta* (bulletin2)] for more general bulletin comparisons.	
Format:	number(8)	External: i8
NA Value:	-999	
Range:	*dnsta* > –999	
Name:	*domain*	
Table:	**datauser**	
Description:	Domain name for a Message Subsystem user	
Format:	varchar2(48)	External: a48
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid internet domain	

Name:	*dprimp*	
Table:	**ex_an**	
Description:	In a comparison of bulletins, *dprimp* is the difference in the number of primary phases between corresponding origins. For a phase to be primary it must be time-defining, a member of the set {P, Pn, Pg, PKP, PKPdf}, and the first arrival at a particular station. The value is [(number-analyst-phases) – (number-expert-phase)] for analyst versus expert-system comparisons and [(number-bulletin1-phases) – (number-bulletin2-phases)] for more general bulletin comparisons.	
Format:	number(8)	External: i8
NA Value:	-999	
Range:	*dprimp* > –999	
Name:	*dropped*	
Table:	qcstats	
Description:	Flag that indicates if the interval was dropped by DFX due to excessive masking	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*dropped* ∈ {0, 1}	
Name:	*drsta*	
Table:	**ex_an**	
Description:	In a comparison of bulletins, *drsta* is the difference in the number of regional non-array stations contributing to corresponding bulletin origins. This value is [*rsta* (analyst) – *rsta* (expert system)] for analyst versus expert-system comparisons or [*rsta* (bulletin1) – *rst*a (bulletin2)] for more general bulletin comparisons	
Format:	number(8)	External: i8
NA Value:	-999	
Range:	*drsta* > -999	
Name:	*dsecondp*	
Table:	**ex_an**	
Description:	In a comparison of bulletins, *dsecondp* is the difference in the number of secondary phases between corresponding bulletin origins. For a phase to be secondary it must be defining and cannot be a member of the phase set {P, Pn, Pg, PKP, PKPdf}. The value is [(number-analyst-phases) – (number-expert-phases)] for analyst versus expert-system comparisons and [(number-bulletin1-phases) – (number-bulletin2-phases)] for more general bulletin comparisons.	
Format:	number(8)	External: i8
NA Value:	-999	
Range:	*dsecondp* > -999	

Name:	*dtime*	
Table:	**bull_comp**, **ex_an**	
Description:	Difference in the origin time between corresponding origins. This value is [*time* (analyst) – *time* (expert system)] for analyst versus expert-system comparisons and [*time* (bulletin1) – *time* (bulletin2)] for more general bulletin comparisons	
Format:	float(24)	External: f8.3
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*dtime* > `-999.0`	
Name:	*dtsta*	
Table:	**ex_an**	
Description:	Difference in the number of teleseismic stations (station/event distance > 2000 km) contributing to the analyst and expert system origins. This value is [*tsta* (analyst) – *tsta* (expert system)] for analyst versus expert-system comparisons and [*tsta* (bulletin1) – *tsta* (bulletin2) for more general bulletin comparisons	
Format:	number(8)	External: i8
NA Value:	`-999`	
Range:	*dtsta* > `-999`	
Name:	*dtype*	
Table:	**origin (origin_ga)**	
Description:	Depth determination flag. This single-character flag indicates the method by which the depth was determined or constrained during the location process. The recommended values are `f` (free), `d` (from depth phases), `r` (restrained by location program) or `g` (restrained by geophysicist). In cases `r` or `g`, either the *auth* column should indicate the agency or person responsible for this action, or the *commid* column should point to an explanation in the **remark** table.	
Format:	varchar2(1)	External: a1
NA Value:	`-` (hyphen)	
Range:	*dtype* ∈ {`f`, `d`, `r`, `g`}	
Name:	*duration*	
Table:	**amplitude**, **chan_groups**, **wfactivity**	
Description:	Duration of the time region for **chan_groups** and **wfactivity**. Total duration of amplitude window for **amplitude**. Combined with *time*, the entire amplitude time window is specified. May also be employed to compute a coda duration magnitude if *amp* and *per* columns contain NA Values.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0` `-999.0` for **amplitude**	
Units:	Seconds	
Range:	*duration* > 0.0, *duration* ≥ `0.0` for **amplitude**	

Name:	*durend*	
Table:	**hydro_arrival**	
Description:	Duration end time	
Format:	float(53)	External: f17.5
NA Value:	`-1.0`	
Units:	Seconds	
Range:	*durend* > 0.0	
Name:	*duronset*	
Table:	**hydro_arrival**	
Description:	Duration onset time	
Format:	float(53)	External: f17.5
NA Value:	`-1.0`	
Units:	Seconds	
Range:	*duronset* > 0.0	
Name:	*edepth*	
Table:	**sitechan**	
Description:	Emplacement depth at which instrument is positioned relative to the value of *elev* in the **site** table	
Format:	float(24)	External: f9.4
NA Value:	NOT ALLOWED	
Units:	Kilometers	
Range:	*edepth* ≥ `0.0`	
Name:	*elev*	
Table:	**site**	
Description:	Surface elevation. This column is the elevation of the surface of the earth above the seismic station (**site**) relative to mean sea level	
Format:	float(24)	External: f9.4
NA Value:	`-999.0`	
Units:	Kilometers	
Range:	`-10.0` ≤ *elev* ≤ `10.0`	
Name:	*ema*	
Table:	**arrival**	
Description:	Emergence angle. This column is the emergence angle of an arrival as observed at a 3-component station or array. The value increases from the vertical direction towards the horizontal.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *ema* ≤ `90.0`	

Name:	*emaillimit*	
Table:	**datauser**	
Description:	Maximum size of message that will be delivered via e-mail in the Message Subsystem	
Format:	number(8)	External: i8
NA Value:	`-1`	
Units:	bytes	
Range:	*emaillimit* ≥ `0`	
Name:	*emailto*	
Table:	**msgdest**	
Description:	Destination email address	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*emares*	
Table:	**assoc (assoc_ga)**	
Description:	Emergence angle residual. This column is the difference between an observed emergence angle and the theoretical prediction for the same phase, assuming an event location as specified by the accompanying *orid*.	
Format:	float(24)	External: f7.1
NA Value:	`-999.0`	
Units:	Degrees	
Range:	`-90.0` ≤ *emares* ≤ `90.0`	
Name:	*endtime*	
Table:	**Affiliation, calibrate**, **bull_comp**, **datadays**, **datacollected, interval**, **missed_class**, **pixdisc**, **qcdata**, **qcstats**, **request**, **sensor**, **wfdisc (wfproto)**	
Description:	Epoch time. Epoch time is given as seconds and fractions of a second since hour 0 January 1, 1970 and stored in a double-precision floating number.	
Format:	float(53)	External: f17.5
NA Value:	+`9999999999.999`	
Units:	Seconds	
Range:	*time* < *endtime* < +9999999999.999	
Name:	*eorid*	
Table:	**ex_an**	
Description:	Expert system origin identifier in an expert system versus analyst origin comparison	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*eorid* > 0	

Name:	*esaz*
Table:	**assoc (assoc_ga)**
Description:	Event-to-station azimuth measured in degrees clockwise from North.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Degrees
Range:	`0.0` ≤ *esaz* < `360.0`

Name:	*etype*
Table:	**origin (origin_ga)**
Description:	An event type that is used to identify the type of seismic event, when known. The recommended event types are:

Value	Description:
`qb`	Quarry blast or mining explosion
`eq`	Earthquake
`e`	Marine explosion
`ex`	Other explosion
`o`	Other source of known origin
`l`	Local event of unknown origin
`r`	Regional event of unknown origin
`t`	Teleseismic event of unknown origin

Format:	varchar2(7) External: a7
NA Value:	- (hyphen)
Range:	Any lowercase character string up to the column size

Name:	*eval_comment*
Table:	**discrimvote**
Description:	Evaluator comments on override
Format:	varchar2(22) External: a22
NA Value:	- (hyphen)
Range:	Any character string up to the column size

Name:	*evid*
Table:	**event, event_control, netmag, origin (origin_ga), request, stamag**
Description:	Event identifier. Each event is assigned a unique positive integer that identifies it in a database. Several records in the **origin** table can have the same *evid*. Analysts have several opinions about the location of the event.
Format:	number(9) External: i9
NA Value:	`-1` NOT ALLOWED for **event**
Range:	*evid* > `0`

Name:	*evname*	
Table:	**event**	
Description:	Event name. This is the common name of the event identified by *evid*.	
Format:	varchar2(32)	External: a32
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	
Name:	*expected*	
Table:	**qcdata**	
Description:	Expected number of seconds of data	
Format:	float(53)	External: f12.1
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*expected* > 0.0	
Name:	*extern_auth*	
Table:	**channame**	
Description:	The external authority using the station name. Not used for translation.	
Format:	varchar2(20)	External: a20
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	
Name:	*extern_chan*	
Table:	**channame**	
Description:	The name of channel as supplied in the data format frame of CD-1 protocol. The name is chosen by the data provider, together with *extern_sta*, which is a primary key	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*extern_sta*	
Table:	**channame**	
Description:	The name of station as supplied in data format frame of CD-1 protocol. The name is chosen by the data provider together with *extern_chan*, which is a primary key.	
Format:	varchar2(6)	External: a6
NA Value:	NOT ALLOWED	
Range:	Any upper-case character string up to the column size	
Name:	*extmsgid*	
Table:	**msgdisc**	
Description:	Value of the *msgid* column in a message that is received by the message system	
Format:	varchar2(20)	External: a20
NA Value:	-1	
Range:	Any character string up to the column size	

Name:	*fileno*	
Table:	**wftape**	
Description:	Tape file number	
Format:	number(4)	External: i4
NA Value:	NOT ALLOWED	
Range:	*fileno* ≥ 0	
Name:	*fileoff*	
Table:	**msgdisc**	
Description:	Number of bytes to the first character of the e-mail file (first character of the e-mail header). F*ileoff* will always be `0` (zero) on the operations system, but will be reset when archived.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Units:	byte	
Range:	*fileoff* > 0	
Name:	*filesize*	
Table:	**msgdisc**	
Description:	Size of file.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Units:	byte	
Range:	*fileoff* > 0	
Name:	*fkqual*	
Table:	**detection**	
Description:	An integer quantifying the quality of the f-k spectrum. An *fkqual* = `1` is high quality and an *fkqual* = `4` is low quality.	
Format:	number(4)	External: i4
NA Value:	`-1`	
Range:	`1`≤ *fkqual* ≤ `4`	
Name:	*flt_rto*	
Table:	**hydro_arrival**	
Description:	Filter ratio	
Format:	float(24)	External: f11.4
NA Value:	`-1.0`	
Range:	*flt_rto* > 0.0	

Name:	*fm*	
Table:	**arrival**	
Description:	First motion. This is a two-character indication of first motion. The first character describes first motion seen on short-period channels and the second holds for long-period instruments. Compression on a short-period sensor is denoted by `c` and dilation by `d`. Compression on a long-period sensor is denoted by `u` and dilation by `r`. Empty character positions will be indicated by dots (for example, `.r` for dilitation on a long-period sensor).	
Format:	varchar2(2)	External: a2
NA Value:	`-` (hyphen)	
Range:	*fm* ∈ all two-letter permutations of `{c , d , .}, {u , r , .}`	

Name:	*foff*	
Table:	**msgdatatype, msgdisc, wfdisc (wfproto)**	
Description:	File offset; the byte offset of a data segment within a physical data file. This column is nonzero if the data reference does not occur at the beginning of the file.	
Format:	number(10)	External: i10
NA Value:	NOT ALLOWED	
Range:	*foff* ≥ `0`	

Name:	*forid*	
Table:	**ex_an**	
Description:	Final origin identifier; the origin identification of the analyst *orid* in an expert system versus analyst origin comparison.	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*forid* > 0	

Name:	*forwardport*	
Table:	**dlman**	
Description:	Forwarder port used by the diskloop manager.	
Format:	number(6)	External: i6
NA Value:	`0`	
Range:	`0` ≤ *forwardport* ≤ `16383`	

Name:	*freq*	
Table:	**apma**	
Description:	Center frequency of the wideband polarization analysis (for example, if only the 2 - 4 Hz and 4 - 8 Hz bands satisfy the signal-to-noise ratio criterion, then *freq* is set to 5.0 Hz)	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Hertz	
Range:	*freq* > 0.0	

Name:	*fstat*	
Table:	**detection**	
Description:	F-statistic; a measure of the signal-to-noise ratio at the peak in the f-k spectrum.	
Format:	float(24)	External: f5.2
NA Value:	`-1.0`	
Range:	*fstat* ≥ `0.0`	
Name:	*ftp_address*	
Table:	**ftpfailed, ftplogin**	
Description:	The ftp address of source site from which attempting to transfer data message.	
Format:	varchar2(64)	External: a64
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*ftpstatus*	
Table:	**ftpfailed**	
Description:	Status of ftp attempt	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	*ftpstatus* ∈ {`retry`, `failed`}	
Name:	*ftype*	
Table:	**interval_files**	
Description:	Indicates the archive file type: ASCII file (`a`), waveform file (`w`), or directory (`d`).	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*ftype* ∈ {`a`, `w`, `d`}	
Name:	*full*	
Table:	**dlfile**	
Description:	File is full (`y`/`n`). Full is set to `y` if the diskloop file is full and `n` if otherwise.	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*full* ∈ {`n`, `y`}	
Name:	*gctp1* through *gctp15*	
Table:	**mapdisc**	
Description:	General cartographic transformation package variable. The default for all values is `0.`	
Format:	float(24)	External: f10.4
NA Value:	NOT ALLOWED	
Range:	See **mapdisc** man page	

Name:	*grn*
Table:	**ev_summary (an_summary, ex_summary), gregion, origin (origin_ga)**
Description:	Geographic region number.
Format:	number(8) External: i8
NA Value:	`-1`
Range:	1 ≤*grn* ≤ 729
Name:	*grname*
Table:	**gregion, pixdisc**
Description:	Geographic region name. This column is the common name of a geographic region. Names may have changed due to changing political circumstances (for example, old `RHODESIA` = new `ZIMBABWE`) (see *grn* and *srname*).
Format:	varchar2(40) External: a40
NA Value:	NOT ALLOWED
Range:	Any character string up to the column size
Name:	*gvhi*
Table:	**ampdescript**
Description:	High group velocity for determining a time window. This column defines the start time of an amplitude measurement window if *toff* is NULL. If *gvhi* is used, then *gvlo* must be used to define the end time of the window.
Format:	float(24) External: f5.2
NA Value:	`-999.0`
Units:	Kilometers/second
Range:	*gvhi* > *gvlo*
Name:	*gvlo*
Table:	**ampdescript**
Description:	Low group velocity for determining a time window. This column defines the end time of an amplitude measurement window if *tlen* is null or if *gvhi* is used to define the start time of the window.
Format:	float(24) External: f5.2
NA Value:	`-999.0`
Units:	Kilometers/second
Range:	*gvlo* > 0.0
Name:	*hamp*
Table:	**amp3c**
Description:	Horizontal amplitude. Absolute maximum amplitude (nm) measured on the root of the sum of the squares of two horizontally-oriented components filtered in a frequency band centered at *cfreq* Hz
Format:	float(24) External: f11.2
NA Value:	`-999.0`
Units:	Nanometers
Range:	*hamp* ≥ `0.0`

Name:	*hang*	
Table:	**sitechan**	
Description:	Horizontal orientation of seismometer. This column specifies the orientation of the seismometer in the horizontal plane, measured clockwise from North. For a North-South orientation with the seismometer pointing toward the North, *hang* = `0.0` For East-West orientation with the seismometer pointing toward the West, *hang* = `270.0` (see *vang*)	
Format:	float(24)	External: f6.1
NA Value:	NOT ALLOWED	
Units:	Degrees	
Range:	`0.0` ≤ *hang* ≤ `360.0`	
Name:	*hmxmn*	
Table:	**apma**	
Description:	Maximum-to-minimum horizontal amplitude ratio defined as $(\lambda_1/\lambda_2)^{1/2}$ where λ_1 and λ_2 are the maximum and minimum eigenvalues obtained by solving the 2-D eigensystem using only the horizontal components. This S-type value is calculated at the time of maximum 3-component amplitude.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Range:	*hmxmn* ≥ `0.0`	
Name:	*hsnr*	
Table:	**amp3c**	
Description:	Horizontal signal-to-noise ratio. Ratio of *hamp* to the root-mean-square amplitude of the root of the sum of the squares of two horizontally oriented components filtered in a frequency band centered at *cfreq* Hz	
Format:	float(24)	External: f10.2
NA Value:	`-999.0`	
Range:	*hsnr* ≥ `0.0`	
Name:	*htov*	
Table:	**amp3c**	
Description:	Horizontal-to-vertical power ratio. One-half times the square of the ratio of *hamp* to *vamp*	
Format:	float(24)	External: f10.2
NA Value:	`-999.0`	
Range:	*htov* ≥ `0.0`	

Name:	*hvrat*	
Table:	**apma**	
Description:	Horizontal-to-vertical power ratio defined as: $(C_3 + C_2)/2C_1$ where C_1, C_2, and C_3 are the diagonal elements of the covariance matrix (C_1 corresponds to the vertical component). This is an S-type value and is calculated at the time of the maximum 3-component amplitude.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Range:	*hvrat* ≥ `0.0`	
Name:	*hvratp*	
Table:	**apma**	
Description:	Horizontal-to-vertical power ratio defined as: $(C_3 + C_2)/2C_1$ where C_1, C_2, and C_3 are the diagonal elements of the covariance matrix (C_1 corresponds to the vertical component). This is a P-type column and is calculated at the time of maximum rectilinearity.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Range:	*hvrat* ≥ `0.0`	
Name:	*hyd_class_code*	
Table:	**hydro_origin**	
Description:	Hydroacoustic event classification	
Format:	number(8)	External: a8
NA Value:	`-999`	
Range:	1 ≤ *hyd_class_code* ≤ 6	
Name:	*hyd_grp_phase*	
Table:	**hydro_arr_group**	
Description:	Hydro-arrival-group phase	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	Any character string up to the column size; currently *hyd_grp_phase* ∈ {`H`, `T`, `N`}	
Name:	*hydro_id*	
Table:	**hydro_arr_group, hydro_assoc**	
Description:	Identifier which is the primary key in the **hydro_arr_group** table and the foreign key in **hydro_assoc**.	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*hydro_id* > 0	

Name:	*hydroloc_code*	
Table:	**hydro_origin**	
Description:	Hydroacoustic origin location code	
Format:	number(8)	External: i8
NA Value:	`-999`	
Range:	*hydroloc_code* ≥ `0`	
Name:	*hydroyield*	
Table:	**hydro_origin**	
Description:	Hydroacoustic estimated yield	
Format:	float(24)	External: f11.2
NA Value:	`-1.0`	
Units:	Kiloton	
Range:	*hydroyield* > 0.0	
Name:	*hydroylderr*	
Table:	**hydro_origin**	
Description:	Hydroacoustic estimated yield error	
Format:	float(24)	External: f11.2
NA Value:	`-1.0`	
Range:	*hydroylderr* > 0.0	
Name:	*id*	
Table:	**ga_tag**	
Description:	Arrival or origin identifier	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*id* > 0	
Name:	*idate*	
Table:	**msgdisc**	
Description:	Initial Julian date that message was received	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	Any valid Julian date	
Name:	*imethod*	
Table:	**msgdisc**	
Description:	Method of transmission for a message	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	*imethod* ∈ {`email`, `ftp`}	

Name:	*inang1*	
Table:	**apma**	
Description:	Apparent incidence angle (measured from the vertical) of the eigenvector (e_1) associated with the largest eigenvalue (λ_1). This column is also called the long-axis incidence angle or the emergence angle. This P-type value is calculated at the time of maximum rectilinearity.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *inang1* < 90.0	

Name:	*inang3*	
Table:	**apma**	
Description:	Apparent incidence angle (measured from the vertical) of the eigenvector (e_3) associated with the smallest eigenvalue (λ_3). This column is also called the short-axis incidence angle. This S-type value is measured at the time of the maximum 3-component amplitude.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *inang3* < 90.0	

Name:	*inarrival*	
Table:	**amplitude**	
Description:	Flag to indicate whether or not *amp* is the same as it is in the **arrival** table	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*inarrival* ∈ {`y`, `n`}	

Name:	*inauth*	
Table:	**wfconv**	
Description:	Flag showing if input data is authenticated	
Format:	varchar2 (1)	External: a1
NA Value:	`-` (hyphen)	
Range:	*inauth* ∈ {`y`, `n`}	

Name:	*incomp*	
Table:	**wfconv**	
Description:	Input data compression type. The only type currently supported is Canadian compression (`CA`).	
Format:	varchar2(2)	External: a2
NA Value:	`-` (hyphen)	
Range:	*incomp* ∈ {`CA`}	

Name:	*inid*	
Table:	**instrument**, **sensor**	
Description:	Instrument identifier. This column is a unique key to the **instrument** table. The *inid* column provides the only link between **sensor** and **instrument**.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED −1 for **sensor**	
Range:	*inid* > 0	
Name:	*inloop*	
Table:	**dlfile**	
Description:	File is part of a diskloop (y/n)	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*inloop* ∈ {n, y}	
Name:	*insamp*	
Table:	**wfconv**	
Description:	Number of input samples per packet	
Format:	number(8)	External: i8
NA Value:	0	
Range:	*insamp* > 0	
Name:	*insname*	
Table:	**instrument**	
Description:	Instrument name. This character string contains the name of the instrument.	
Format:	varchar2(50)	External: a50
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*instant*	
Table:	**sensor**	
Description:	Snapshot indicator. When instant = y, the snapshot was taken at the time of a discrete procedural change, such as an adjustment of the instrument gain; when instant = n, the snapshot is of a continuously changing process, such as calibration drift. This value is important for tracking time corrections and calibrations. The default value is y.	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*instant* ∈ {y, n}	

Name:	*instype*	
Table:	**instrument**, **wfdisc (wfproto)**	
Description:	Instrument type. This character string is used to indicate the instrument type (for example, `SRO`, `ASRO`, `DWWSSN`, `LRSM`, and `S-750`).	
Format:	varchar2(6)	External: a6
NA Value:	- (hyphen)	
Range:	Any upper-case character string up to the column size	
Name:	*intern_chan*	
Table:	**channame**	
Description:	Name of channel chosen by data consumer. The translation is from *extern_chan* to *intern_chan*.	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any lower-case character string up to the column size	
Name:	*intern_chanid*	
Table:	**channame**	
Description:	The *chanid* corresponding to the *intern_sta*, *intern_chan*. It is a foreign key into the **sitechan** table.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*intern_chanid* > 0	
Name:	*intern_sta*	
Table:	**channame**	
Description:	The name of the station as chosen by the data consumer. The translation is from *extern_sta* to *intern_sta*.	
Format:	varchar2(6)	External: a6
NA Value:	NOT ALLOWED	
Range:	Any upper-case character string up to the column size	
Name:	*intid*	
Table:	**msgdisc**	
Description:	Internal identifier for message tracking	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*intid* > 0	
Name:	*intidtype*	
Table:	**msgdisc**	
Description:	Identifier type for the *intid*.	
Format:	varchar2(16)	External: a16
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that is a valid identifier in the schema	

Name:	*intvlid*	
Table:	**interval**, **interval_files**	
Description:	Interval identifier. Each interval is assigned a unique positive integer that identifies it in the database	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*intvlid* > 0	
Name:	*intype*	
Table:	**wfconv**	
Description:	Input fixed width datatype	
Format:	varchar2(2)	External: a2
NA Value:	- (hyphen)	
Range:	same as datatype	
Name:	*inwfactivity*	
Table:	**chan_groups**	
Description:	Indicates whether or not this *class*/*name*/*duration* will appear in the **wfactivity** table	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	*inwfactivity* ∈ {0 , 1}	
Name:	*iphase*	
Table:	**arrival**	
Description:	Reported phase. This eight-character column holds the name initially given to a seismic phase. Standard seismological labels for the types of signals (or phases) are used (for example, P, PKP, PcP, pP). Both upper- and lower-case letters are available and should be used when appropriate [for example, pP or PcP (see *phase*)].	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that conforms to seismological practice	
Name:	*isrc*	
Table:	**msgdisc**	
Description:	Initial source of message.	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*itime*	
Table:	**msgdest**, **msgdisc**	
Description:	Initial time when message was received	
Format:	float(53)	External: f17.5
NA Value:	-999.0	
Units:	Seconds	
Range:	*itime* > 0.0	

Name:	*jdate*	
Table:	**arrival**, **datadays**, **detection**, **origin (origin_ga)**, **pixdisc**, **qcdata**, **qcstats**, **sensor**, **wfdisc (wfproto)**	
Description:	Julian date. Date of an arrival, origin, seismic recording, etc. The same information is available in epoch time, but the Julian date format is more convenient for many types of searches. Dates B.C. are negative. The year will never equal 0000, and the day will never equal 000. Where only the year is known, the day of the year is 001; where only year and month are known, the day of year is the first day of the month. Only the year is negated for B.C., so 1 January of 10 B.C. is `0010001` (see *time*).	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	Julian dates are of the form yyyyddd; must be consistent with the accompanying time column	
Name:	*kbscause*	
Table:	**ex_an**	
Description:	Obsolete column	
Format:	varchar2(7)	External: a7
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*keyname*	
Table:	**lastid**	
Description:	Identifier type. This column contains the actual name of a key whose last assigned numeric value is saved in *keyvalue*. Typical values are of the form `<xx>id`.	
Format:	varchar2(15)	External: a15
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*keyvalue*	
Table:	**lastid**	
Description:	Current identifier value. This column maintains the last assigned value (a positive integer) of the counter for the specified *keyname*. The keyvalue is the last counter value used for the column *keyname*. Key values are maintained in the database to ensure uniqueness.	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*keyvalue* > 0	
Name:	*label*	
Table:	**mapdisc**	
Description:	Header for *Map* listing. A label, such as world, categorizes each *Map*. *Label* is used to build a sorted list of maps in the *Map* program.	
Format:	varchar2(65)	External: a65
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*last_mig_date*	
Table:	**mig_date**	
Description:	Last date of database migration	
Format:	date	External: a17
NA Value:	NOT ALLOWED	
Range:	Any valid ORACLE date	
Name:	*lastfailedtime*	
Table:	**ftpfailed**	
Description:	Time of most recent attempt to retrieve data message by ftp	
Format:	float(53)	External: i4
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*lastfailedtime* > -9999999999.999	
Name:	*lat*	
Table:	**aoi**, **mappoint**, **origin (origin_ga)**, **site**	
Description:	Geographic latitude. Locations North of the equator have positive latitudes	
Format:	float(53)	External: f11.6
NA Value:	NOT ALLOWED for **aoi** `-999.0`	
Units:	Degrees	
Range:	`-90.0` ≤ *lat* ≤ `90.0`	
Name:	*latmajor*	
Table:	**mapdisc**	
Description:	Latitude interval for displaying major grid lines in the *Map* application	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	0.0 < *latmajor* < 90.0	
Name:	*latminor*	
Table:	**mapdisc**	
Description:	Latitude interval for displaying minor grid lines in the *Map* application	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	0.0 < *latminor* < 90.0	

Name:	*latorigradians*	
Table:	**mapdisc**	
Description:	Latitude origin radians. Coordinates in radians of the lower left corner in the *Map* application. The m*ap* application uses this for mercator projections only.	
Format:	float(24)	External: f9.4
NA Value:	`-999.0`	
Units:	Radians	
Range:	$-\pi/2 <$ *latorigradians* $< \pi/2$	
Name:	*lddate*	
Table:	**affiliation (stanet)**, **alphasite**, **amp3c**, **ampdescript**, **amplitude**, **aoi**, **apma**, **arrival, assoc (assoc_ga)**, **bull_comp**, **calibrate**, **chan_groups**, **channame**, **colordisc**, **datadays**, **datacollected**, **datauser**, **detection**, **discrimuse**, **discrimvote**, **dlfile**, **dlman**, **ev_summary (an_summary**, **ex_summary)**, **event**, **event_control**, **ex_an**, **ftpfailed**, **ftplogin**, **gregion**, **hydro_arr_group**, **hydro_arrival**, **hydro_assoc**, **hydro_origin**, **instrument**, **interval**, **interval_files**, **lastid**, **mapcolor**, **mapdisc**, **mapover**, **mappoint**, **mig_date**, **mig_rules**, **missed_class**, **msgaux**, **msgdatatype**, **msgdest**, **msgdisc**, **netmag**, **network**, **qcstats**, **origerr (origerr_ga)**, **origin (origin_ga)**, **overlaydisc**, **pixdisc**, **qcdata**, **remark**, **request**, **sensor**, **site**, **siteaux**, **sitechan**, **sregion**, **stamag**, **station_hist**, **station_type**, **timestamp**, **wfactivity**, **wfconv**, **wfdisc (wfproto)** , **wftag**, **xtag**	
Description:	Load date. Date and time the record was inserted into the database. For the **bull_comp** table, *lddate* is the date of the comparison.	
Format:	date	External: a19
NA Value:	NOT ALLOWED	
Range:	Any valid ORACLE date	
Name:	*length*	
Table:	**dlfile**	
Description:	Length of file, bytes (**dlfile**), waveform length, and bytes (**wfaux**)	
Format:	number(10)	External: i10
NA Value:	NOT ALLOWED	
Units:	Bytes	
Range:	*length* > 0	
Name:	*lineno*	
Table:	**remark**	
Description:	Line number. This integer is assigned as a sequence number for multiple line comments.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*lineno* > 0	

Name:	*loc_all_stas*	
Table:	**event_control**	
Description:	Logical descriptor that informs the location process whether it should only use stations with source-dependent corrections in event locations. If TRUE (`1`), use all stations in event location. If FALSE (`0`), only use phase data from stations possessing either an source-specific station corrections (SSSC), source-region station timing (SRST), or test-site correction. Any data without a valid correction will not be included in the final event location. Only meaningful if *src_dpnt_corr* is > `0.0`. Default is TRUE (`1`).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	*loc_all_stas* ∈ {`0`, `1`}	
Name:	*loc_alpha_only*	
Table	**event_control**	
Description:	Logical descriptor that restricts phase data to be used in event location to only those stations contained in the substation list. If FALSE (`0`), use all stations provided in the **site** table. If TRUE (`1`), only PRIMARY station data is used to locate events. This is option desirable in cases where the station network has varying station qualities contributing to events. Default is FALSE (`0`).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	*loc_alpha_only* ∈ {`0`, `1`}	
Name:	*loc_dist_varwgt*	
Table:	**event_control**	
Description:	Logical descriptor that informs the location process if predefined distance variance weighting should be applied to the event location. The predefined weighting is a set of data variances as a function of distance. If FALSE (`0`) and both *user_var_wgt* and *srst_var_wgt* are also set to FALSE (`0`), then variances are determined by the *deltim*, *delslo*, and *delaz* from **arrival**. Default is FALSE (`0`).	
Format:	number(1)	External: i1
NA Value:	NOT ALLOWED	
Range:	*loc_dist_varwgt* ∈ {`0`, `1`}	
Name:	*loc_sdv_mult*	
Table:	**event_control**	
Description:	Large residual multiplier factor. This column is only meaningful when *loc_sdv_screen* is set to TRUE (`1`). If *loc_sdv_screen* is TRUE (`1`), all data with travel-time/azimuth/slowness residuals greater than this factor times its data variance (standard error) will be ignored during any given iteration of the location process. Default is `3.0`.	
Format:	float(24)	External: f5.2
NA Value:	NOT ALLOWED	
Range:	*loc_sdv_mult* > 0.0	

Name:	*loc_sdv_screen*
Table:	**event_control**
Description:	Logical descriptor that tells the location process whether or not to ignore data with travel-time/azimuth/slowness residuals greater than *loc_sdv_mult* times its data standard error in determining an event location. If FALSE (`0`), include data regardless of its residuals, provided it meets other pertinent conditions. Default is FALSE (`0`).
Format:	number(1) External: i1
NA Value:	NOT ALLOWED
Range:	*loc_sdv_screen* ∈ {`0`, `1`}
Name:	*loc_src_dpnt_reg*
Table:	**event_control**
Description:	Source-dependent region identifier for event location. If source-dependent corrections are applied as part of event location process (*src_dpnt_corr* > `0`), then *loc_src_dpnt_reg* indicates region name.
Format:	varchar2(15) External: a15
NA Value:	- (hyphen)
Range:	Any character string up to the column size
Name:	*location*
Table:	**interval_files**
Description:	Location code for the file. This single word providing a location code for the file.
Format:	varchar2(20) External: a20
NA Value:	- (hyphen)
Range:	*location* ∈ {longterm , permanent}
Name:	*logat*
Table:	**arrival**
Description:	Log of amplitude divided by period. This measurement of signal size is often reported instead of the amplitude and period separately. This column is only filled if the separate measurements are not available.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Log (nanometers/seconds)
Range:	*logat* > 0.0
Name:	*lon*
Table:	**aoi**, **mappoint, origin**, **site**
Description:	Geographic longitude. Llongitudes are measured positive East of the Greenwich meridian.
Format:	float(53) External: f11.6
NA Value:	NOT ALLOWED for **aoi** `-999.0`
Units:	Degrees
Range:	`-180.0` ≤ *lon* ≤ `180.0`

Name:	*lonmajor*	
Table:	**mapdisc**	
Description:	Longitude interval (in degrees) for displaying major grid lines in the *Map* application	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	0.0 < *lonmajor* < 180.0	
Name:	*lonminor*	
Table:	**mapdisc**	
Description:	Longitude interval (in degrees) for displaying minor grid lines in the *Map* application	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	–180.0 < *lonminor* < 180.0	
Name:	*lonorigradians*	
Table:	**mapdisc**	
Description:	Longitude origin radians. Coordinates in radians of the lower left corner in the *Map* application. *Map* uses this for mercator projections only.	
Format:	float(24)	External: f9.4
NA Value:	`-999.0`	
Units:	Radians	
Range:	$-\pi \leq latorigradians \leq \pi$	
Name:	*lsta*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of local arrival times associated with an event. `Local` is currently defined as a station-event distance of less than 250 km	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*lsta* > 0	
Name:	*machine*	
Table:	**dlfile, dlman**	
Description:	Fully qualified domain name of the computer where the connection manager resides	
Format:	varchar2(32)	External: a32
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that points to a valid machine	

Name:	*mag_all_stas*
Table:	**event_control**
Description:	Logical descriptor that informs magnitude process whether or not it should only use amplitude information from stations with magnitude test-site corrections. If TRUE (`1`), use all amplitude information in event magnitudes. If FALSE (`0`), use only amplitude data from stations possessing a magnitude test-site correction, any data without a valid correction will not be included in the magnitude determination. Only meaningful if *mag_test_site* is NOT NULL or `-`. Default is TRUE (`1`).
Format:	number(1) External: i1
NA Value:	NOT ALLOWED
Range:	*mag_all_stas* ∈ {`0`, `1`}
Name:	*mag_alpha_only*
Table:	**event_control**
Description:	Logical descriptor that restricts amplitude data to be used in the magnitude determination to only those stations contained in the substation list. If FALSE (`0`), use all valid amplitudes. If TRUE (`1`), only primary seismic station data is used for the magnitude calculation. This option is desirable in cases where the station network has varying station qualities contributing to the magnitude. Default is FALSE (`0`).
Format:	number(1) External: i1
NA Value:	NOT ALLOWED
Range:	*mag_alpha_only* ∈ {`0`, `1`}
Name:	*mag_sdv_mult*
Table:	**event_control**
Description:	Magnitude large residual multiplier factor; meaningful only when *mag_sdv_screen* is set to TRUE (`1`). If *mag_sdv_screen* is TRUE (`1`), all amplitude with magnitude residuals greater than this factor times its data variance (standard error) will be ignored by the magnitude process. Default is `3.0`.
Format:	float(24) External: f5.2
NA Value:	NOT ALLOWED
Range:	*mag_sdv_mult* > 0.0
Name:	*mag_sdv_screen*
Table:	**event_control**
Description::	Logical descriptor that tells magnitude process whether or not to ignore amplitude data with magnitude residuals greater than *mag_sdv_mult* times its data standard error in determining the given magnitude. If FALSE (`0`), include data regardless of its residuals provided it meets other pertinent conditions. Default is FALSE (`0`).
Format:	number(1) External: i1
NA Value:	NOT ALLOWED
Range:	*mag_sdv_screen* ∈ {`0`, `1`}

Name:	*mag_src_dpnt_reg*	
Table:	**event_control**	
Description::	Source-dependent region identifier for magnitude determination. If source-dependent corrections are applied as part of the event magnitude determination process, then *mag_src_dpnt_reg* indicates the region name.	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*magdef*	
Table:	**stamag**	
Description::	Magnitude defining switch. This one-character flag indicating whether or not a station magnitude for a given **stamag** record was used in determining the network magnitude. This column is defining (*magdef* = d) if it is used in network magnitude calculation or nondefining (*magdef* = n) if it is not used.	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*magdef* ∈ {d, n}	
Name:	*magid*	
Table:	**netmag, stamag**	
Description:	Network magnitude identifier. This value is assigned to identify a network magnitude in the **netmag** table. This column is required for every network magnitude. Magnitudes given in **origin (origin_ga)** must reference a network magnitude with *magid* = *mbid*, *mlid* or *msid*, whichever is appropriate (see *mbid*, *mlid*, or *msid*).	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*magid* > 0	
Name:	*magnitude*	
Table:	**netmag, stamag**	
Description:	Magnitude. This column gives the magnitude value of the type indicated in *magtype*. The value is derived in a variety of ways, which are not necessarily linked directly to an arrival (see *magtype*, *mb*, *ml*, and *ms*).	
Format:	float(24)	External: f7.2
NA Value:	NOT ALLOWED -999.0 for **netmag**	
Units:	Magnitude	
Range:	-9.99 <*magnitude* < 50.00	

Name:	*magres*	
Table:	**stamag**	
Description:	Magnitude residual. Difference between the magnitude for a given **stamag** record and network magnitude	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Units:	Magnitude	
Range:	-10.0 < *magres* < 10.0	
Name:	*magtype*	
Table:	**netmag, stamag**	
Description:	Magnitude type (for example, *mb*)	
Format:	varchar2(6)	External: a6
NA Value:	NOT ALLOWED	
Range:	Any magnitude type up to the column size	
Name:	*mapfiletype*	
Table:	**mapdisc**	
Description:	Specifies how the *Map* program handles the referenced *Map* file. If *mapfiletype* = `all`, then the program reads the file in its entirety. If *mapfiletype* = `blk`, then the program reads only the blocks necessary for the display area.	
Format:	varchar2(4)	External: a4
NA Value:	NOT ALLOWED	
Range:	*mapfiletype* ∈ {`all`, `blk`}	
Name:	*mapid*	
Table:	**mapcolor, mapdisc, mapover**	
Description:	Mapdisc identifier. Each **mapdisc** is assigned a unique positive integer that identifies it in a database.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*mapid* > 0	
Name:	*mapname*	
Table:	**mapdisc**	
Description:	Name of the map. Each map in the *Map* application is assigned a name for identifying the map in a list of all maps.	
Format:	varchar2(64)	External: a64
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	

Name:	*maptype*	
Table:	**mapdisc**	
Description:	Type of map. A positive integer enumerator for identifying the output graphic type, either raster or vector (*maptype* = `1` for raster and *maptype* = `2` for vector)	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*maptype* ∈ `{1, 2}`	
Name:	*masked*	
Table:	**qcdata**, **qcstats**	
Description:	Number of seconds in the processing interval masked due to point-spikes, spikes, or constant value segments	
Format:	float(53)	External: f17.5
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*masked* ≥ `0.0`	
Name:	*masks*	
Table:	**qcdata**	
Description:	Number of masks	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*masks* ≥ `0`	
Name:	*max_endtime*	
Table:	**wfactivity**	
Description:	The maximum endtime value found in the **wfdisc (wfproto)** table for that time region	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*max_endtime* > -9999999999.999	
Name:	*mb*	
Table:	**origin (origin_ga)**	
Description:	Body wave magnitude, m_b [**origin, origin_ga)**]. This is the body wave magnitude of an event. The identifier *mbid* that points to *magid* in the **netmag** table is associated with this column. The information in that record summarizes the method of analysis and data used (see *magnitude*, *magtype*, *ml*, and *ms*).	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Units:	Magnitude	
Range:	-9.99 < *mb* < 50.00	

Name:	*mb_max_dist*	
Table:	**event_control**	
Description:	Body wave magnitude (m_b). Station magnitudes at distances less than *mb_min_dist* will not be used in network magnitude calculations.	
Format:	float(24)	External: f9.4
NA Value:	`-999.0`	
Units:	Degrees	
Range:	`0.0` ≤ *mb_min_dist* ≤ `180.0`	
Name:	*mb_min_dist*	
Table:	**event_control**	
Description:	Body wave magnitude (m_b). Station magnitudes at distances less than *mb_min_dist* will not be used in network magnitude calculations.	
Format:	float(24)	External: f9.4
NA Value:	`-999.0`	
Units:	Degrees	
Range:	`0.0` ≤ *mb_min_dist* ≤ `180.0`	
Name:	*mbid*	
Table:	**origin (origin_ga)**	
Description:	Magnitude identifier for mb. This attribute stores the *magid* for a record in **netmag**. The identifier *mbid* is a foreign key joining **origin** to **netmag** where **origin**.*mbid* = **netmag**.*magid* (see *magid*, *mlid*, and *msid*).	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*mbid* > 0	
Name:	*mean_az*	
Table:	**station_hist**	
Description:	Mean azimuth	
Format:	float(53)	External: f8.3
NA Value:	NOT ALLOWED	
Range:	`0.0` ≤ *mean_az* < 360.0	
Name:	*mean_slow*	
Table:	**station_hist**	
Description:	Mean slowness	
Format:	float(53)	External: f8.3
NA Value:	NOT ALLOWED	
Range:	*mean_slow* > 0.0	

Name:	*mean_time*	
Table:	**station_hist**	
Description:	Arrival time mean	
Format:	float(53)	External: f8.3
NA Value:	NOT ALLOWED	
Range:	*mean_time* > 0.0	
Name:	*mfoff*	
Table:	**msgdisc**	
Description:	Offset in bytes to beginning of message	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*mfoff* > 0	
Name:	*min_time*	
Table:	**wfactivity**	
Description:	The minimum time value found in the **wfdisc (wfproto)** table for that time region	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*min_time* > -9999999999.999	
Name:	*missing*	
Table:	**qcstats**	
Description:	Number of seconds missing data in the interval	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	0.0 ≤ *missing* < 9999999999.99999	
Name:	*ml*	
Table:	**origin (origin_ga)**	
Description:	Local magnitude (M_L) of an event. The identifier *mlid*, which points to *magid* in the **netmag** tables, is associated with this column. The information in that record summarizes the method of analysis and the data used (see *magnitude*, m*a*gtype, *mb*, and *ms*).	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Units:	Magnitude	
Range:	-9.99 < *ml* < 50.00	

Name:	*mlid*	
Table:	**origin (origin_ga)**	
Description:	Magnitude identifier for local magnitude (M_L). This attribute stores the *magid* for a record in **netmag**. The identifier *mlid* is a foreign key joining **origin (origin_ga)** to **netmag**, where **table.***mlid* = **table**.*magid* (see *magid*, *msid*, and *mbid*).	
Format:	number(9)	External: i9
NA Value:	-1	
Range:	*mlid* > 0	
Name:	*mmodel*	
Table:	**event_control, stamag**	
Description:	Magnitude model. This character string identifies the magnitude model employed for station (**stamag**) or overall network magnitude calculation. In **stamag**, *mmodel* is the unique magnitude model as extracted from the magnitude correction file. In **event_control**, *mmodel* indicates only whether or not mixed models were employed (mixed) or a unique magnitude model was used for all stations. In the latter case, it would be identical to **stamag**.*mmodel*.	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*modauthor*	
Table:	**request**	
Description:	Author of last state change	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*moddate*	
Table:	**interval**, **wfactivity**	
Description:	Modification date. Date and time the record last updated (*state* column) in the database	
Format:	date	External: a17
NA Value:	NOT ALLOWED	
Range:	Any valid ORACLE date	
Name:	*modtime*	
Table:	**request**	
Description:	Modification time. The epoch time that the record last updated in the database	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*modtime* > -9999999999.999	

Name:	*mpdescrip*	
Table:	**mappoint**	
Description:	Arbitrary string describing the referenced geographic point	
Format:	varchar2(50)	External: a50
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*mplabel*	
Table:	**mappoint**	
Description:	String used as a label for the geographic point described by a record in the **mappoint** table (for example, `Paris`, `London`, `K8`, etc.).	
Format:	varchar2(65)	External: a65
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*mptype*	
Table:	**mappoint**	
Description:	String specifying the type of geographic point described by a record in the **mappoint** table (examples include cities, mines, etc.)	
Format:	varchar2(20)	External: a20
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*ms*	
Table:	**origin**	
Description:	This is the surface wave magnitude for an **event**. The identifier *msid*, which points to *magid* in the **netmag** table, is associated with this column. The information in that record summarizes the method of analysis and the data used (see *magnitude*, *magtype*, *mb*, and *ml*).	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Range:	-9.99 < *ms* < 50.00	
Name:	*msgdformat*	
Table:	**msgdatatype**	
Description:	General format of the data that follows	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*msgdid*	
Table:	**msgdest**	
Description:	Message destination identifier	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*msgdid* > 0	

Name:	*msgdtype*	
Table:	**msgdatatype**	
Description:	Data type of a data section within a message	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a recognized data type	
Name:	*msgid*	
Table:	**msgaux, msgdatatype, msgdest, msgdisc, ftpfailed**	
Description:	Message identifier. In **msgdest**, this column is the message identifier of the response message.	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*msgid* > 0	
Name:	*msgrow*	
Table:	**msgaux**	
Description:	Number of lines in a message	
Format:	number(4)	External: i4
NA Value:	`-1`	
Range:	*msgrow* > 0	
Name:	*msgsrc*	
Table:	**msgdisc**	
Description:	Message source identifier	
Format:	varchar2(16)	External: a16
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	
Name:	*msgstatus*	
Table:	**msgdatatype, msgdest, msgdisc**	
Description:	Status of message for **msgdest**, **msgdisc**; for **msgdatatype**, status of data section	
Format:	varchar2(32)	External: a32
NA Value:	– (hyphen)	
Range:	Any character string up to the column size for **msgdest** *msgstatus* ∈ {`DONE`, `FAILED`} for **msgdatatype and msgdisc**	
Name:	*msgtype*	
Table:	**datauser, msgdisc**	
Description:	Message type	
Format:	varchar2(16)	External: a16
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	

Name:	*msgver*	
Table:	**msgdisc**	
Description:	Message Subsystem version number	
Format:	varchar2(8)	External: a8
NA Value:	– (hyphen)	
Range:	Any character string up to the column size	

Name:	*msid*	
Table:	**origin (origin_ga)**	
Description:	Magnitude identifier for *ms*. This column stores the *magid* for a record in **netmag**. The identifier *msid* is a foreign key joining **origin** to **netmag,** where **table**.*msid* = **table**.*magid* (see *magid*, *mlid*, and *mbid*).	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*msid* > 0	

Name:	*msize*	
Table:	**msgdatatype, msgdisc**	
Description:	Size of bytes of message or section of message	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*msize* > 0	

Name:	*mtype*	
Table:	**ampdescript**	
Description:	Measurement type. This column defines how the amplitude is measured in a given time window. The following values are allowed: `peak` (maximum amplitude), `stav` (maximum short-term average amplitude), `rms` (root-mean-squared amplitude), `peak2tr` (maximum peak-to-trough amplitude), and `1stpeak` (first motion amplitude).	
Format:	varchar2(8)	External: a8
NA Value:	– (hyphen)	
Range:	*mtype* ∈ {`peak`, `stav`, `rms`, `peak2tr`, `1stpeak`}	

Name:	*multev*	
Table:	**ex_an**	
Description:	Indicates whether or not another analyst event solution is within 50 km and 5 minutes of the analyst event (for example, multiple event)	
Format:	varchar2(4)	External: a4
NA Value:	– (hyphen)	
Range:	*multev* ∈ {`y`, `n`}	

Name:	*name*	
Table:	**chan_groups**, **interval**, **wfactivity**	
Description:	Name of an interval.	
Format:	varchar2(20)	External: a20
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*narr1*	
Table:	**bull_comp**	
Description:	Number of associated arrivals for *orid1*	
Format:	Number(9)	External: i9
NA Value:	- (hyphen)	
Range:	*narr1* ≥ 0	
Name:	*narr2*	
Table:	**bull_comp**	
Description:	Number of associated arrivals for *orid2*	
Format:	number(9)	External: i9
NA Value:	-1	
Range:	*narr2* ≥ 0	
Name:	*nass*	
Table:	**origin (origin_ga)**	
Description:	Number of associated arrivals. This column gives the number of arrivals associated with the origin	
Format:	number(4)	External: i4
NA Value:	-1	
Range:	*nass* > 0	
Name:	*ncalib*	
Table:	**instrument**	
Description:	Nominal calibration factor. This conversion factor maps digital data to earth displacement. The factor holds true at the oscillation period specified by *ncalper*. A positive value means ground motion increasing in component direction (up, North, East) is indicated by increasing counts. A negative value means the opposite. Actual calibration for a particular recording is determined using the **wfdisc (wfproto)** and **sensor** tables (see *calratio*).	
Format:	float(24)	External: f16.6
NA Value:	NOT ALLOWED	
Units:	Nanometers/digital count	
Range:	*ncalib* ≠ 0.0	

Name:	*ncalper*	
Table:	**instrument**	
Description:	Calibration period. This column is the period for which *ncalib* is valid	
Format:	float(24)	External: f16.6
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*ncalper* > 0.0	
Name:	*nchans*	
Table:	**qcdata**	
Description:	Number of channels	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*nchans* > 1	
Name:	*ncomp*	
Table:	**station_type**	
Description:	Number of components	
Format:	number(8)	External: i8
NA Value:	`-999`	
Range:	*ncomp* ∈ {1 , 3}	
Name:	*nconstseg*	
Table:	**qcstats**	
Description:	Number of constant valued segments in the detection processing interval	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*nconstseg* ≥ 0	
Name:	*ndef*	
Table:	**ev_summary (an_summary, ex_summary)**, **origin (origin_ga)**	
Description:	Number of time-defining phases	
Format:	number(4)	External: i4
NA Value:	`-1`	
Range:	0 < *ndef* ≤ nass	
Name:	*ndef1*	
Table:	**bull_comp**	
Description:	Number of time-defining phases for *orid1*	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*ndef1* ≥ 0	

Name:	*ndef1arr2*	
Table:	**bull_comp**	
Description:	Number of defining arrivals for *orid1* that are arrivals (defining or nondefining) for *orid2*	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*ndef1arr2* ≥ `0`	
Name:	*ndef2*	
Table:	**bull_comp**	
Description:	Number of time-defining phases for *orid2*	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*ndef2* ≥ `0`	
Name:	*ndef2arr1*	
Table:	**bull_comp**	
Description:	Number of defining arrivals for *orid2* that are arrivals (defining or nondefining) for *orid1*	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*ndef2arr1* ≥ `0`	
Name:	*ndp*	
Table:	**origin (origin_ga)**	
Description:	Number of depth phases. This column gives the number of depth phases used in calculating *depth*/*depdp* (see *depdp*)	
Format:	number(4)	External: i4
NA Value:	`-1`	
Range:	*ndp* ≥ `0`	
Name:	*nearaz*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Azimuth from nearest station to the event	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *nearaz* < 360.0	
Name:	*neardist*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Distance from the event to the closest station	
Format:	float(24)	External: f8.3
NA Value:	`-1.0`	
Units:	Kilometers	
Range:	*neardist* > 0.0	

Name:	*nearsta*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Code for the nearest station to the event	
Format:	varchar2(6)	External: a6
NA Value:	- (hyphen)	
Range:	Any character string that is a valid station code	
Name:	*net*	
Table:	**affiliation (stanet), hydro_arr_group, netmag, network**	
Description:	Unique network identifier. This character string is the name of a seismic network (for example, `WWSSN`).	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED - (hyphen) for **netmag**	
Range:	Any character string up to the column size	
Name:	*netname*	
Table:	**network**	
Description:	Network name. This character string contains the name of a network.	
Format:	varchar2(80)	External: a80
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*nettype*	
Table:	**network**	
Description:	Network type. This four-character string specifies the type of network [array (`ar`), local area (`lo`), world-wide (`ww`) for the given value of *net*]	
Format:	varchar2(4)	External: a4
NA Value:	- (hyphen)	
Range:	Any lower-case character string up to the column size	
Name:	*new_endtime*	
Table:	**wfaudit**	
Description:	The *new_endtime* value provided to the **wfdisc (wfproto)** trigger	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*new_endtime* > -9999999999.999	

Name:	*new_time*	
Table:	**wfaudit**	
Description:	The *new_time* value provided to the **wfdisc (wfproto)** trigger	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*new_time* > -9999999999.999	
Name:	*nhydarr*	
Table:	**hydro_arr_group**	
Description:	Number of arrivals in the group	
Format:	number(4)	External: i4
NA Value:	-1	
Range:	*nhydarr* ≤ number of stations in the group	
Name:	*nmatch*	
Table:	**bull_comp**	
Description:	Number of matching arrivals (defining or nondefining) between *orid1* and *orid2*	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*nmatch* ≥ 0	
Name:	*nois*	
Table:	**siteaux**	
Description:	Nominal background seismic noise level	
Format:	float(24)	External: f10.1
NA Value:	-1.0	
Units:	Nanometers	
Range:	*nois* ≥ 0.0	
Name:	*noise*	
Table:	**qcdata**	
Description:	Average noise amplitude	
Format:	float(24)	External: f8.3
NA Value:	-1.0	
Range:	*noise* ≥ 0.0	
Name:	*noissd*	
Table:	**siteaux**	
Description:	Standard deviation of the log noise amplitude	
Format:	float(24)	External: f5.2
NA Value:	-999.0	
Range:	*noissd* > 0.0	

Name:	*normamp*	
Table:	**hydro_arrival**	
Description:	Normalized hydroacoustic amplitude for computing yield	
Format:	float(24)	External: f11.4
Units:	`-1.0`	
NA Value:	dB reference one microvolt	
Range:	0.0 < *normamp* < 100.0	
Name:	*nsamp*	
Table:	**wfdisc (wfproto)**	
Description:	Number of samples. This quantity is the number of samples in a waveform segment.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*nsamp* > 0	
Name:	*nseg*	
Table:	**qcstats**	
Description:	Number of masked segments in the detection processing interval	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*nseg* ≥ `0`	
Name:	*nsta*	
Table:	**ev_summary (an_summary, ex_summary), missed_class, netmag**	
Description:	Number of stations. In **ev_summary (an_summary, ex_summary)**, this column is the number of stations with an associated arrival. In **netmag**, this column is the number of stations contributing to the network magnitude estimate.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*nsta* > 0	
Name:	*num_in_series*	
Table:	**hydro_origin**	
Description:	Number of event in hydroacoustic series of events [for example, event `10529` is *num_in_series* = `23` of 60 events in its series (see *serid*)]	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*num_in_series* > 0	
Name:	*numfailedattempt*	
Table:	**ftpfailed**	
Description:	Number of failed attempts to retrieve a message via ftp	
Format:	number(4)	External: i4
NA Value:	`0`	
Range:	*numfailedattempt* > 0	

Name:	*objtype*	
Table:	**ga_tag**	
Description:	Defines the id as either an *arid* (a) or an *orid* (o)	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*objtype* ∈ {a, o}	
Name:	*offdate*	
Table:	**aoi, chan_groups**, **site**, **sitechan**	
Description:	Turn off date. This column is the Julian Date on which the station or sensor indicated was turned off, dismantled, or moved (see *ondate*)	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	Julian date of the form yyyyddd	
Name:	*old_endtime*	
Table:	**wfaudit**	
Description:	The *old_endtime* value provided to the **wfdisc (wfproto)** trigger	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*old_endtime* > -9999999999.999	
Name:	*old_time*	
Table:	**wfaudit**	
Description:	The *old_time* value provided to the **wfdisc (wfproto)** trigger	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*old_time* > -9999999999.999	
Name:	*ondate*	
Table:	**aoi, chan_groups**, **site**, **sitechan**	
Description:	Turn on date. Date on which the archive specifications, regional coefficient, station, sensor, or subscription indicated became applicable or began operating. The columns *offdate* and *ondate* are not intended to accommodate temporary downtimes, but rather to indicate the time period for which the columns of the station (*lat*, *lon*, *elev*) are valid for the given station code. Stations are often moved, but with the station code remaining unchanged.	
Format:	number(8)	External: I8
NA Value:	NOT ALLOWED	
Range:	Julian date of the form yyyyddd	

Name:	*onset_time*	
Table:	**hydro_arrival**	
Description:	Estimated onset time of signal	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*onset_time* > -9999999999.999	
Name:	*orid*	
Table:	**assoc (assoc_ga)**, **discrimuse**, **discrimvote**, **ev_summary (an_summary, ex_summary)**, **event_control**, **hydro_origin**, **missed_class, netmag**, **origin (origin_ga)**, **origerr (origerr_ga)**, **request**, **stamag**	
Description:	Origin identifier that relates a record in these tables to a record in the **origin (origin_ga)** table	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*orid* > `0`	
Name:	*orid1*	
Table:	**bull_comp**	
Description:	Origin identifier from the bulletin1 database **origin (origin_ga)** table	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*orid1* > 0	
Name:	*orid2*	
Table:	**bull_comp**	
Description:	Origin identifier from the bulletin2 database **origin (origin_ga)** table	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*orid2* > 0	
Name:	*outauth*	
Table:	**wfconv**	
Description:	Flag showing if output data is authenticated	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*outauth* ∈ {y, n}	
Name:	*outcomp*	
Table:	**wfconv**	
Description:	Output data compression type. The only type currently supported is Canadian compression (CA).	
Format:	varchar2(2)	External: a2
NA Value:	`0`	
Range:	*outcomp* ∈ {CA}	

Name:	*outsamp*	
Table:	**wfconv**	
Description:	Number of output samples per packet	
Format:	number(8)	External: i8
NA Value:	0	
Range:	*outsamp* > 0	
Name:	*outtype*	
Table:	**wfconv**	
Description:	Output fixed width datatype	
Format:	varchar2(2)	External: a2
NA Value:	- (hyphen)	
Range:	same as datatype	
Name:	*overlayid*	
Table:	**mapover, overlaydisc**	
Description:	Overlay identifier	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*overlayid* > 0	
Name:	*overlayname*	
Table:	**overlaydisc**	
Description:	Name of the map overlay	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*override*	
Table:	**discrimvote**	
Description:	Evaluator override vote	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	*override* ≥ 0	
Name:	*Parid*	
Table:	**Amplitude**	
Description:	Predicted arrival identifier	
Format:	number(9)	External: i9
NA Value:	-1	
Range:	*parid* > 0	

Name:	*partition*	
Table:	**dlfile**	
Description:	Disk partition name	
Format:	varchar2(64)	External: a64
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid disk partition	
Name:	*password*	
Table:	**ftplogin**	
Description:	Password for remote ftp site for pushing data messages from the local site	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*per*	
Table:	**amplitude**, **arrival**	
Description:	Measured period at the time of the amplitude measurement	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*per* > 0.0	
Name:	*phase*	
Table:	**apma**, **assoc (assoc_ga)**, **stamag**	
Description:	Phase type. The identity of a phase that has been associated to an arrival. Standard labels for phases are used (for example, `P`, `PKP`, `PcP`, `pP`, etc.). Both upper- and lower-case letters are available and should be used when appropriate (for example, `pP` or `PcP`).	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen) if this column does not apply to seismic phases	
Range:	Any character string up to the column size that conforms to scientific practice	
Name:	*pixid*	
Table:	**pixdisc**	
Description:	Picture identifier	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*pixid* > `0`	
Name:	*pixdescr*	
Table:	**pixdisc**	
Description:	Description of image	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*planlr*	
Table:	**apma**	
Description:	Planarity of an S-type polarization column defined as $1 - l_3/l_2$, where l_2 and l_3 are eigenvalues from the decomposition of the covariance matrix. Planarity is measured at the time of maximum 3-component amplitude.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Range:	`0.0` ≤ *planlr* ≤ `1.0`	
Name:	*plans*	
Table:	**apma**	
Description:	Planarity of an S-type polarization column defined as $1 - l_3/l_2$, where l_2 and l_3 are eigenvalues from the decomposition of the covariance matrix. Planarity is measured at the time of maximum 3-component amplitude. The only difference between *plans* and *planlr* is in the definition of the overlapping time windows.	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Range:	`0.0` ≤ *plans* ≤ `1.0`	
Name:	*pmdescr*	
Table:	**datadays, station_hist**	
Description:	Description of *PerfMon* processing state.	
Format:	varchar2(64)	External: a64
NA Value:	`-` (hyphen)	
Range:	Any character string up to the column size	
Name:	*pocid*	
Table:	**datauser**	
Description:	Point of contact identifier	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*pocid* > `0`	
Name:	*pointspike*	
Table:	**qcstats**	
Description:	Amount of data in the detection processing interval masked due to point-spikes	
Format:	float(53)	External: f17.5
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*pointspike* ≥ `0.0`	

Name:	*pphasetime*	
Table:	**apma**	
Description:	Epoch time at which P-type polarization columns are estimated. This value is the center of the time window with maximum rectilinearity	
Format:	float(53)	External: f17.5
NA Value:	`-9999999999.999`	
Units:	Seconds	
Range:	*pphasetime* > 0.0	
Name:	*prefdlid*	
Table:	**alphasite**	
Description:	The *dlid* preferred *DLMan* for a given station	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*prefdlid* > 0	
Name:	*prefer_loc*	
Table:	**event_control**	
Description:	Location identifier. This column indicates which of three possible location solutions is the location for the given origin. The hypocenter can be either held to a surface location (`S`), determined with no constraints at all (free depth, `F`), or restrained based on the settings of *constrain_ot*, *constrain_latlon*, and *constrain_depth*, (`R`). The constrained location (`R`) can be fixed in origin time/latitude and longitude/depth. When *prefer_loc* indicates a surface (`S`) or free depth (`F`) location, *prefer_loc* takes precedence to the actual constraint settings of *constrain_ot*, *constrain_latlon* and *constrain_depth*. Default is `S`.	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*prefer*_loc ∈ {`F`, `S`, `R`}	
Name:	*prefor*	
Table:	**event**	
Description:	Preferred origin. This column holds the origin identifier (*orid*) that points to the preferred origin for a seismic **event**	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*prefor* > `0`	
Name:	*prefport*	
Table:	**alphasite**	
Description:	Preferred network port for a given station	
Format:	number(6)	External: i8
NA Value:	`0`	
Range:	`1` ≤ *prefport* ≤ `16383`	

Name:	*primp*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of primary time-defining phases. A primary phase is defined as the first phase for a given station belonging to the set (`P`, `Pn`, `Pg`, `PKP`, `PKPdf`).	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*primp* ≥ `0`	
Name:	*priority*	
Table:	**datauser**	
Description:	Priority assigned to process	
Format:	number(2)	External: i2
NA Value:	NOT ALLOWED	
Range:	*priority* > `0`	
Name:	*procclass*	
Table:	**mig_date, timestamp**	
Description:	Process class used to group processes	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size (upper-case for **mig_date**)	
Name:	*processing*	
Table:	**datacollected, pixdisc**	
Description:	*PerfMon* mode of processing	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid processing mode	
Name:	*process_state*	
Table:	**ga_tag**	
Description:	Processing state	
Format:	varchar2(20)	External: a20
NA Value:	NOT ALLOWED	
Range:	A set of strings defined at each installation for each automated processing system, *process_state* ∈ {`aa_processed`, `analyst_reviewed`, `assoc_first`, `driver_restricted`, `locked_association`, `probdet_restricted`, `requested`, `wc_restricted`}	
Name:	*procname*	
Table:	**mig_date, timestamp**	
Description:	Process name that identifies a process within a process class.	
Format:	varchar2(16)	External: a16
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size (upper-case for **mig_date**)	

Name:	*projection*	
Table:	**mapdisc**	
Description:	Projection of the *Map*. A positive integer enumerator for uniquely classifying the stereographic projection of the *Map*. Azimuthal equidistant = `2`; Mercator = `3`	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*projection* ∈ {2, 3}	
Name:	*ptmcor*	
Table:	**siteaux**	
Description:	P-wave arrival time correction	
Format:	float(24)	External: f6.3
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*ptmcor* > –999.0	
Name:	*pub_access*	
Table:	**pixdisc**	
Description:	Sets access permissions on images created by *PerfMon.*	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*pub_access* ∈ {0, 1}	
Name:	*qcdataid*	
Table:	**qcdata**	
Description:	QC data identifier	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*qcdataid* > 0	
Name:	*qcstatsid*	
Table:	**qcstats**	
Description:	Data quality statistics identifier	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*qcstatsid* > 0.0	

Name:	*qual*	
Table:	**arrival**	
Description:	Onset quality. This single -character flag is used to denote the sharpness of the onset of a seismic phase. This relates to the timing accuracy as follows: `i` (impulsive) – accurate to ±0.2 seconds `e` (emergent) – accuracy between ±(0.2 to 1.0 seconds) `w` (weak) – timing uncertain to > 1 second.	
Format:	varchar (1)	External: a1
NA Value:	`-` (hyphen)	
Range:	*qual* ∈ {`i`, `e`, `w`, `1`, `2`, `3`, `4`}	
Name:	*quer_seq_no*	
Table:	**mig_rules**	
Description:	Order of this part of query for data migration	
Format:	number(4)	External: I4
NA Value:	NOT ALLOWED	
Range:	*quer_seq_no* ≥ `1`	
Name:	*query_type*	
Table:	**mig_rules**	
Description:	Type of query for data migration	
Format:	varchar2(20)	External: a20
NA Value:	NOT ALLOWED	
Range:	*query_type* ∈ {`COMMIT`, `CREATE_TABLE`, `DELETE`, `DROP_TABLE`, `SELECT_INSERT`}	
Name:	*rdepthp*	
Table:	**ex_an**	
Description:	Number of depth phases renamed by the analyst (see *ddepthp* for definition of depth phase)	
Format:	number(8)	External: i8
NA Value:	`-999`	
Range:	*rdepthp* > `0`	
Name:	*reaptime*	
Table:	**dlfile**	
Description:	Time at which the contents of the disk loop file may be discarded and the file re-used	
Format:	float(53)	External: f17.5
NA Value:	`0.0`	
Units:	Seconds	
Range:	*reaptime* > 0.0	

Name:	*reason*	
Table:	**discard**	
Description:	Reason why automated system event was discarded by an analyst	
Format:	varchar2(30)	External: a30
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*rect*	
Table:	**apma, arrival**	
Description:	Signal rectilinearity defined as: $1 - (l_3 + l_2)/2l_1$ where l_1, l_2, and l_3 are the three eigenvalues from the decomposition of the covariance matrix. This value is the maximum rectilinearity for all overlapping time windows	
Format:	float(24)	External: f7.3
NA Value:	`-1.0`	
Range:	`0.0` ≤ *rect* ≤ `1.0`	
Name:	*refaz*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Azimuth to nearest reference point	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *refaz* < 360.0	
Name:	*refdist*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Distance to nearest reference point	
Format:	float(24)	External: f8.3
NA Value:	`-1.0`	
Units:	Kilometers	
Range:	*refdist* ≥ `0.0`	
Name:	*refid*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Reference location identifier	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*refid* > 0	

Name:	*reflat*	
Table:	**mapdisc**	
Description:	Latitude reference. Latitude of the center of the *Map* application's projection (used for azimuthal equidistant projections only).	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	`-90.0` ≤ *reflat* ≤ `90.0`	
Name:	*reflon*	
Table:	**mapdisc**	
Description:	Longitude reference. Longitude of the center of the *Map* application's projection (used for azimuthal equidistant projections only).	
Format:	float(53)	External: f11.6
NA Value:	`-999.0`	
Units:	Degrees	
Range:	`-180.0` ≤ *reflon* ≤ `180.0`	
Name:	*refoffsetlat*	
Table:	**mapdisc**	
Description:	Latitude offset reference. This value is the reference (in pixels) from the lower left corner of the map to the center of the *Map* application's projection. In the case where the reference point is at the center of the map, the offsets are equal to half the map width and height (used for azimuthal equidistant projections only).	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Pixels	
Range:	*refoffsetlat* > 0.0	
Name:	*refoffsetlon*	
Table:	**mapdisc**	
Description:	Longitude offset reference. This value is the reference (in pixels) from the lower left corner of the map to the center of the *Map* application's projection (used for azimuthal equidistant projections only).	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Pixels	
Range:	*refoffsetlon* > 0.0	
Name:	*refsta*	
Table:	**site**	
Description:	Reference station. This string specifies the reference station with respect to which array members are located (see *deast*, *dnorth*).	
Format:	varchar2(6)	External: a6
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*region*
Table:	**datacollected**
Description:	Limits of geographic region. Minimum/maximum latitudes and longitudes
Format:	varchar2(30) External: a30
NA Value:	NOT ALLOWED
Range:	Any character string up to the column size that is a valid region

Name:	*rely*
Table:	**siteaux**
Description:	Station reliability. This column is an estimate of the percentage of time that the station is up
Format:	float(24) External: f5.2
NA Value:	`-1.0`
Range:	`0.0` ≤ *rely* ≤ `1.0`

Name:	*remark*
Table:	**remark**
Description:	Descriptive text. This single line of text is an arbitrary comment about a record in the database. The comment is linked to its parent table only by forward reference from *commid* in the record of the table of interest (see *commid*, *lineno*).
Format:	varchar2(80) External: a80
NA Value:	- (hyphen)
Range:	Any character string up to the column size

Name:	*reqid*
Table:	**request**
Description:	Request identifier. Unique key to allow tracking of requests by the Message Subsystem
Format:	number(9) External: i9
NA Value:	NOT ALLOWED
Range:	*reqid* > 0

Name:	*requestor*
Table:	**request**
Description:	Original requestor of this data. The requestor is the person or program that requests this waveform data
Format:	varchar2(15) External: a15
NA Value:	- (hyphen)
Range:	Any character string up to the column size

Name:	*req_state*
Table:	**request**
Description:	Current request state
Format:	varchar2(16) External: a16
NA Value:	- (hyphen)
Range:	Any character string up to the column size

Name:	*retime*	
Table:	**ex_an**	
Description:	Number of phases re-timed by an analyst	
Format:	number(8)	External: I8
NA Value:	`-1`	
Range:	*retime* ≥ `0`	
Name:	*retrieved*	
Table:	**qcdata**	
Description:	Actual number of seconds of data	
Format:	float(24)	External: f12.1
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	*retrieved* > 0.0	
Name:	*rid*	
Table:	**amp3c**	
Description:	Recipe identifier	
Format:	varchar2(8)	External: a8
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*rms*	
Table:	**hydro_arrival**	
Description:	Bubble pulse amplitude versus *rms*	
Format:	float(24)	External: f11.4
NA Value:	`-1.0`	
Range:	*rms* > 0.0	
Name:	*rotation*	
Table:	**mapdisc**	
Description:	Map rotation. This is the rotation of the projection from 0˚, or due North. Rotation specifies the azimuth of the y-raster in degrees clockwise from north (for azimuthal equidistant projections only).	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Degrees	
Range:	`0.0` ≤ *rotation* < 360.0	
Name:	*rprimp*	
Table:	**ex_an**	
Description:	Number of primary phases renamed by the analyst (see *dprimp* for definition of primary phase)	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*rprimp* ≥ `0`	

Name:	*rsecondp*	
Table:	**ex_an**	
Description:	Number of secondary phases renamed by the analyst (see *dsecondp* for definition of secondary phase).	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*rsecondp* ≥ `0`	
Name:	*rsptype*	
Table:	**instrument**	
Description:	Instrument response type. This value denotes the style in which detailed calibration data is stored. The neighboring column *dfile* tells where the calibration data is saved. *rsptype* = `paz` indicates the data is the poles and zeroes of the Laplace transform *rsptype* = `fap` indicates the data is amplitude/phase values at a range of frequencies *rsptype* = `fir` indicates that the response type is a finite impulse response table r*sptype* = `pazfir` indicates a combination of poles, zeros, and finite impulse response Other codes may be defined.	
Format:	varchar2(6)	External: a6
NA Value:	NOT ALLOWED	
Range:	Any lower-case character string up to the column size	
Name:	*rsta*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of nonarray regional arrival times. Regional is defined as a station-event distance not less than 250 km and up to 2000 km.	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*rsta* ≥ `0`	
Name:	*rt*	
Table:	**hydro_arrival**	
Description:	Ratio of the magnitude of the largest positive peak after the zero lag point on the auto covariance function (*r*(`t`)) to the magnitude of the auto covariance function at the zero lag point (*r*(`0`))	
Format:	float(24)	External: f11.4
NA Value:	`-1.0`	
Range:	0.0 < *rt* < 1.0	
Name:	*running*	
Table:	**dlman**	
Description:	Flag indicating whether or not *DLMan* is running.	
Format:	varchar2(1)	External: a1
NA Value:	– (hyphen)	
Range:	*running* ∈ {`y`, `n`}	

Name:	*samprate*	
Table:	**instrument, wfdisc (wfproto)**	
Description:	Sampling rate. This column is the sample rate in samples per second. In the **instrument** table, the column value is specifically the nominal sample rate, not accounting for clock drift. In **wfdisc (wfproto)**, the value may vary slightly from the nominal to reflect clock drift.	
Format:	float(24)	External: f11.7
NA Value:	NOT ALLOWED	
Units:	1/second	
Range:	*samprate* > 0.0	
Name:	*scale*	
Table:	**mapdisc**	
Description:	Map scale.	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Radians per pixel for mercator projections; km per pixel for azimuthal equidistant projections	
Range:	*scale* > 0.0	
Name:	*sd_az*	
Table:	**station_hist**	
Description:	Azimuth residual	
Format:	float(53)	External: f6.1
NA Value:	NOT ALLOWED	
Range:	`0.0` ≤ *sd_az* < 360.0	
Name:	*sd_slo*	
Table:	**station_hist**	
Description:	Slowness residual	
Format:	float(53)	External: f6.3
NA Value:	NOT ALLOWED	
Range:	*sd_slo* > 0.0	
Name:	*sd_time*	
Table:	**station_hist**	
Description:	Arrival time residual	
Format:	float(53)	External: f6.3
NA Value:	NOT ALLOWED	
Range:	*sd_time* > 0.0	

Name:	*sdepth*
Table:	**origerr (origerr_ga)**
Description:	Depth error. This is the maximum error of a depth estimate for a level of confidence given by *conf* (see *smajax*, *sminax*, and *sxx*, *syy*, *szz*, *stt*, *sxy*, *sxz*, *syz*, *stx*, *sty*, *stz*)
Format:	float(24) External: f9.4
NA Value:	`-1.0`
Units:	Kilometers
Range:	*sdepth* > 0.0
Name:	*sdobs*
Table:	**origerr (origerr_ga)**
Description:	Standard error of one observation. This column is derived from the discrepancies in the arrival times of the phases used to locate an event. This column is defined as the square root of the sum of the squares of the time residuals divided by the number of degrees of freedom. The latter is the number of defining observations [*ndef* in **origin (origin_ga)**] minus the dimension of the system solved (`4` if depth is allowed to be a free variable, `3` if depth is constrained).
Format:	float(24) External: f9.4
NA Value:	`-1.0`
Range:	*sdobs* > 0.0
Name:	*seaz*
Table:	**assoc (assoc_ga)**, **detection**
Description:	Station-to-event azimuth calculated from the station and event locations and measured clockwise from North.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Degrees
Range:	`0.0` ≤ *seaz* ≤ `360.0`
Name:	*seazlr*
Table:	**apma**
Description:	Azimuth of the eigenvector (e_3) associated with the smallest eigenvalue (λ_3). It is corrected by 180˚ to give an estimate of the station-to-event azimuth (with an 180˚ ambiguity). It is an S-type column calculated at the time of the maximum 3-component amplitude. The only difference between *seazs* and *seazlr* is in the definition of the overlapping time windows.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Degrees
Range:	`0.0` ≤ *seazlr* ≤ `360.0`

Name:	*seazp*
Table:	**apma**
Description:	Azimuth of the eigenvector (e_1) associated with the largest eigenvalue (λ_1). This value is corrected by 180˚ to give an estimate of the station-to-event azimuth. This P-type value is calculated at the time of maximum rectilinearity.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Degrees
Range:	`0.0` ≤ *seazp* ≤ `360.0`

Name:	*seazs*
Table:	**apma**
Description:	Azimuth of the eigenvector (e_3) associated with the smallest eigenvalue (λ_3). This value corrected by 180˚ to give an estimate of the station-to-event azimuth (with an 180˚ ambiguity). This column is an S-type column calculated at the time of the maximum 3-component amplitude. The only difference between *seazs* and *seazlr* is in the definition of the overlapping time windows.
Format:	float(24) External: f7.2
NA Value:	`-999.0`
Units:	Degrees
Range:	`0.0` ≤ *seazs* ≤ `360.0`

Name:	*secondp*
Table:	**ev_summary (an_summary, ex_summary)**
Description:	Number of time-defining secondary phases. A secondary phase is any phase not in the set (`P`, `Pn`, `Ps`, `PkP`, `PKPdf`).
Format:	number(8) External: i8
NA Value:	`-1`
Range:	*secondp* ≥ `0`

Name:	*segtype*
Table:	**wfdisc (wfproto)**
Description:	Segment type. This column indicates if a waveform is `o` (original), `v` (virtual), `s` (segmented), or `d` (duplicate)
Format:	varchar2(1) External: a1
NA Value:	- (hyphen)
Range:	*segtype* ∈ {`o`, `v`, `s`, `d`}

Name:	*seismic_geochar*
Table:	**aoi**
Description:	Seismic geographic region characteristic. There are four characteristics that describe the geographic region in which an event is located. This type specifies whether the event is located in an area in which has historically been the site of infrequent [aseismic (`a`)] or regular [seismic (`s`)] occurrences of natural earthquakes (see *aoi_geochar*, *depth_geochar*, and *terrain_geochar*).
Format:	varchar2(1) External: a1
NA Value:	`NOT ALLOWED`
Range:	*seismic_geochar* ∈ {`a` | `s`}

Name:	*seq_contents*	
Table:	**mig_rules**	
Description:	Contents of a query for data migration	
Format:	varchar2(200)	External: a200
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid SQL query	
Name:	*seq_type*	
Table:	**mig_rules**	
Description:	Type of sequence to be added to the query	
Format:	varchar2(15)	External: a15
NA Value:	NOT ALLOWED	
Range:	*seq_type* ∈ {`END_LDDATE`, `START_LDDATE`, `TEXT`}	
Name:	*serid*	
Table:	**hydro_origin**	
Description:	Series identifier. The series identifies is a unique positive number assigned to a hydroacoustic set of events which occur in a series. Each series has a unique number or *serid* associated with it (see *num_in_series*).	
Format:	number(8)	External: i8
NA Value:	`-1`	
Range:	*serid* > 0	
Name:	*servicetime*	
Table:	**datauser**	
Description:	Last time a request from the user with the *userid* in the **datauser** table was serviced	
Format:	float(53)	External: f17.5
NA Value:	`-999999999.999`	
Units:	Seconds	
Range:	*servicetime* ≥ `0.0`	
Name:	*sigdet*	
Table:	**missed_class**	
Description:	Indicates number of arrivals detected in both bulletins	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*sigdet* ≥ `0`	
Name:	*sigtype*	
Table:	**msgdisc**	
Description:	Digital signature type	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	

Name:	*slodef*	
Table:	**assoc (assoc_ga)**	
Description:	Slowness defining code. This one-character flag indicates whether or not the slowness of a phase was used to constrain the event location. This column is defining (*slodef* = `d`) or nondefining (*slodef* = `n`) for this arrival.	
Format:	varchar2(1)	External: a1
NA Value:	`-` (hyphen)	
Range:	*slodef* ∈ {`d`, `n`}	
Name:	*slores*	
Table:	**assoc (assoc_ga)**	
Description:	Slowness residual. This column gives the difference between an observed slowness and a theoretical prediction. The prediction is calculated for the related phase and event origin described in the record.	
Format:	float(24)	External: f7.2
NA Value:	`-999.0`	
Units:	Seconds/degree	
Range:	*slores* > –999.0	
Name:	*slotid*	
Table:	**wfaudit**	
Description:	Each row in **wfaudit** has a unique *slotid* value	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*slotid* > 0	
Name:	*slow*	
Table:	**arrival, detection, hydro_arr_group**	
Description:	Observed slowness of a detected arrival	
Format:	float(24)	External: f7.2
NA Value:	`-1.0`	
Units:	Seconds/kilometers for **detection, hydro_arr_group** Seconds/degree in **arrival** table	
Range:	*slow* ≥ `0.0`	
Name:	*smajax*	
Table:	**origerr (origerr_ga)**	
Description:	Semi-major axis of error ellipse for a given confidence. This value is the length of the semi-major axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane. The level of confidence is specified by *conf* (see *sdepth*, *sminax*, and *sxx*, *syy*, *szz*, *stt*, *sxy*, *sxz*, *syz*, *stx*, *sty*, *stz*).	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Kilometers	
Range:	*smajax* > 0.0	

Name:	*sminax*	
Table:	**origerr (origerr_ga)**	
Description:	Semi-minor axis of error ellipse. This value is the length of the semi-minor axis of the location error ellipse. The value is found by projecting the covariance matrix onto the horizontal plane. The level of confidence is specified by *conf* (see *sdepth*, *smajax*, and *sxx*, *syy*, *szz*, *stt*, *sxy*, *sxz*, *syz*, *stx*, *sty*, *stz*).	
Format:	float(24)	External: f9.4
NA Value:	`-1.0`	
Units:	Kilometers	
Range:	*sminax* > 0.0	
Name:	*snr*	
Table:	**amplitude, apma, arrival, detection**	
Description:	Signal-to-noise ratio. This is an estimate of the ratio of the amplitude of the signal to amplitude of the noise immediately preceding it. For **apma**, this value is based on the maximum 3-component amplitudes (see *amps*). This column is the average signal-to-noise ratio for the frequency bands that contributed to the final polarization estimates.	
Format:	float(24)	External: f10.2
NA Value:	`-1.0`	
Range:	*snr* > 0.0	
Name:	*snthrsh*	
Table:	**siteaux**	
Description:	Nominal signal-to-noise ratio	
Format:	float(24)	External: f5.2
NA Value:	`-1.0`	
Range:	*snthrsh* > 1.0	
Name:	*sphasetime*	
Table:	**apma**	
Description:	Epoch time at which S-type polarization columns are estimated. This is the center of the time window with the maximum 3-component ampliude	
Format:	float(53)	External: f17.5
NA Value:	`-9999999999.999`	
Units:	Seconds	
Range:	*sphasetime* > 0.0	
Name:	*spike*	
Table:	**qcstats**	
Description:	Amount of data in detection processing interval masked due to spikes	
Format:	float(53)	External: f17.5
NA Value:	`-999.0`	
Units:	Seconds	
Range:	*spike* ≥ `0.0`	

Name:	*splitev*	
Table:	**ex_an**	
Description:	Indicates whether or not the analyst event solution contains arrivals that were previously associated with two or more expert system events	
Format:	varchar2(4)	External: a4
NA Value:	– (hyphen)	
Range:	*splitev* ∈ {y, n}	
Name:	*sproid*	
Table:	**detection**	
Description:	Uniquely identifies a set of parameters used in the signal processing.	
Format:	number(8)	External: i8
NA Value:	–1	
Range:	*sproid* > 0	
Name:	*src*	
Table:	**mig_rules**	
Description:	Source database for migration.	
Format:	varchar2(10)	External: a10
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid name of a database server	
Name:	*src_dpnt_corr*	
Table:	**event_control**	
Description:	Identifies whether or not and what type of source-dependent corrections were applied to the location: 0 = No source-dependent corrections applied to the event location 1 = Test-site travel-time corrections applied to the event location 2 = Source-region station-timing (SRST) corrections applied to the event location 3 = Regional level source-specific station corrections (SSSC) applied to the event location. SRST correction is not applied, even if it exists 4 = Local level SSSCs applied to the location. SRST correction is not applied, even if it exists.	
Format:	number(2)	External: i2
NA Value:	NOT ALLOWED	
Range:	*src_dpnt_corr* ∈ {0, 1, 2, 3, 4}	
Name:	*src_tbl*	
Table:	**mig_rules**	
Description:	Source table for database migration	
Format:	varchar2(30)	External: a30
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size that is a valid table name	

Name:	*srn*	
Table:	**origin (origin_ga), sregion**	
Description:	Seismic region number (see *grn*, *grname*, and *srname*)	
Format:	number(8)	External: i8
NA Value:	-1	
Range:	$1 \leq srn \leq 50$	
Name:	*srname*	
Table:	**sregion**	
Description:	Seismic region name. This column is the common name of a seismic region. Names may have changed due to changing political circumstances (see *srn* and *grname*).	
Format:	varchar2(40)	External: a40
NA Value:	NOT ALLOWED	
Range:	Any upper-case character string up to the column size	
Name:	*sta*	
Table:	**affiliation (stanet), alphasite, arrival, assoc (assoc_ga), calibrate, chan_groups, detection, discrimuse, dlfile, hydro_arrival, qcdata, qcstats, request, sensor, site, siteaux, sitechan, stamag, station_hist, station_type, wfconv, wfdisc (wfproto)**	
Description:	Station code. This is the code name of a seismic observatory and identifies a geographic location recorded in the site table	
Format:	varchar2(6)	External: a6
NA Value:	NOT ALLOWED	
Range:	Any upper-case character string up to the column size	
Name:	*staname*	
Table:	**site**	
Description:	Station name/Description:. This value is the full name of the station whose codename is in *sta* [for example, one record in the site table connects *sta* = ANMO to *staname* = ALBUQUERQUE, NEW MEXICO (SRO)].	
Format:	varchar2(50)	External: a50
NA Value:	- (hyphen)	
Range:	Any upper-case character string up to the column size	
Name:	*staper*	
Table:	**siteaux**	
Description:	Standard period at which noise estimates are made	
Format:	float(24)	External: f5.2
NA Value:	-1.0	
Units:	Seconds	
Range:	$staper > 0.0$	

Name:	*stassid*	
Table:	**arrival**	
Description:	Identification of a group of arrivals from the same station originating from the same event	
Format:	number(9)	External: i9
NA Value:	`-1`	
Range:	*stassid* > 0	
Name:	*state*	
Table:	**interval**	
Description:	The processing state of the interval within the automated processing system	
Format:	varchar2(16)	External: a16
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*statecount*	
Table:	**msgaux, request**	
Description:	Count of failures. When *state* = failed, the *statecount* column records the number of failures to acquire this data.	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*statecount* ≥ 0	
Name:	*statype*	
Table:	**site, station_type**	
Description:	Station type;character string specifies the station type. Recommended entries are single station (`ss`) or array (`ar`).	
Format:	varchar2(4)	External: a4
NA Value:	- (hyphen)	
Range:	*statype* ∈ {`ss`, `ar`}	
Name:	*stav*	
Table:	**detection**	
Description:	Short-term average used to describe the amplitude of a signal. The amplitude is averaged over a small time interval, typically 1 - 2 seconds.	
Format:	float(24)	External: f11.5
NA Value:	–1.0	
Units:	Nanometers	
Range:	*stav* > 0.0	

Name:	*stdconstval*	
Table:	**qcstats**	
Description:	Standard deviation of data in masked constant segments	
Format:	float(53)	External: f17.5
NA Value:	–999.0	
Units:	Same as waveform data	
Range:	*stdconstval* ≥ `0.0`	
Name:	*stime*	
Table:	**origerr (origerr_ga)**	
Description:	Origin time error. This column denotes the time uncertainty that accompanies the average error ellipse location (see *smajax*, *sminax*, and *sdepth*).	
Format:	float(24)	External: f6.3
NA Value:	–1.0	
Units:	Seconds	
Range:	*stime* ≥ 0.0	
Name:	*stmcor*	
Table:	**siteaux**	
Description:	S-wave arrival time correction.	
Format:	float(24)	External: f6.3
NA Value:	–999.0	
Units:	Seconds	
Range:	*stmcor* > –999.0	
Name:	*strike*	
Table:	**origerr (origerr_ga)**	
Description:	Strike of major axis of error ellipse. This column is the strike of the semi-major axis of the location error ellipse, measured in degrees clockwise from the North (see *smajax*).	
Format:	float(24)	External: f6.2
NA Value:	–1.0	
Units:	Degrees	
Range:	0.0 ≤ *strike* ≤ 360.0	
Name:	*strip*	
Table:	**wfconv**	
Description:	Flag showing whether or not the data is stripped of headers (`y/n`).	
Format:	varchar2(1)	External: a1
NA Value:	– (hyphen)	
Range:	*strip* ∈ {`n`, `y`}	

Name:	*stype*	
Table:	**arrival**	
Description:	Signal type. This single-character flag indicates the event or signal type. The following definitions hold: `l` = Local event `r` = Regional event `t` = Teleseismic event `m` = Mixed or multiple event `g` = Glitch (for example, non-seismic detection) `e` = Calibration activity obfuscated the data `l`, `r`, and `t` Supplied by the reporting station or as an output of post-detection processing `g` and `e` Come from analyst comment or from status bits from GDSN and RSTN data	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*stype* ∈ {`l`, `r`, `t`, `m`, `g`, `e`}	
Name:	*sub_status*	
Table:	**msgaux**	
Description:	Cause of failure	
Format:	varchar2(24)	External: a24
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*subject*	
Table:	**msgdisc**	
Description:	Subject header from an e-mail message	
Format:	varchar2(64)	External: a64
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*subtopic*	
Table:	**pixdisc**	
Description:	Subtype of data processing	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*subtype*	
Table:	**msgdisc, interval_files**	
Description:	Specification of whether or not the request includes waveforms. In the future, this column may contain indications of other message subtypes.	
Format:	varchar2(2)	External: a2
NA Value:	- (hyphen)	
Range:	*subtype* ∈ {V, R, L}	

Name:	*sxx*, *syy*, *szz*, *stt*, *sxy*, *sxz*, *syz*, *stx*, *sty*, *stz*	
Table:	**origerr (origerr_ga)**	
Description:	Elements of the covariance matrix for the location identified by *orid*. The covariance matrix is symmetric (and positive definite) so that *sxy* = *syx*, and so on, (`x`, `y`, `z`, `t`) refer to latitude, longitude, depth, and origin time, respectively. These columns (together with *sdobs*, *ndef*, and *dtype*) provide the information necessary to construct the K-dimensional (K = 2, 3, 4) confidence ellipse or ellipsoids at any confidence limit desired.	
Format:	float(24)	External: f15.4
NA Value:	`-1.0`	
Units:	*sxx*, *syy*, *szz*, *sxy*, *szx*, *syz* – kilometers squared (km^2) *stt* – seconds squared (sec^2) *stx*, *sty*, *stz* – kilometers per second (km/sec)	
Range:	*sxx*, *syy*, *szz*, *stt* > 0.0	
Name:	*tagid*	
Table:	**wftag**	
Description:	Tagname value. This column contains the value of a foreign key identified in tagname [for example, if *tagname* is `arid`, then **wftag** may be joined to arrival where **arrival**.*arid* = **wftag**.*tagid*. If *tagname* is `orid`, then **wftag** and **origin (origin_ga)** may be joined where **origin**.*orid* = **wftag**.*tagid*.]	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*tagid* > 0	
Name:	*tagname*	
Table:	**wftag**	
Description:	Tagname type. This value is the name of the foreign key whose value is in *tagid*	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	*tagname* ∈ {`arid`, `evid`, `orid`, `stassid`, `msgid`}	
Name:	*tapename*	
Table:	**wftape**	
Description:	Name of tape volume.	
Format:	varchar2(32)	External: a32
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column string	
Name:	*task_num*	
Table:	**mig_rules**	
Description:	Order of this migration task	
Format:	number(4)	External: i4
NA Value:	NOT ALLOWED	
Range:	*task_num* ≥ 1	

Name:	*telep*	
Table:	**missed_class**	
Description:	Indicates the number of teleseismic p phases in an event	
Format:	number(8)	External: i8
NA Value:	NOT ALLOWED	
Range:	*telep* ≥ 0	
Name:	*termination_time*	
Table:	**hydro_arrival**	
Description:	Estimated termination time of signal	
Format:	float(53)	External: f17.5
NA Value:	-9999999999.999	
Units:	Seconds	
Range:	*termination_time* > -9999999999.999	
Name:	*terrain_geochar*	
Table:	**aoi**	
Description:	Terrain geographic region characteristic. There are four characteristics that describe the geographic region in which an event is located. This type specifies whether the event is located in a landmass (l) or in a body of water such as the ocean, sea, or lake (o) (see *aoi_geochar*, *depth_geochar*, and *seismic_geochar*).	
Format:	varchar2(1)	External: a1
NA Value:	NOT ALLOWED	
Range:	*terrain_geochar* ∈ {l | o}	
Name:	*thatdb*	
Table:	**xtag**	
Description:	Database account for the records specified by *thatname* and *thatid*.	
Format:	varchar2(32)	External: a32
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that is a valid account name	
Name:	*thatid*	
Table:	**xtag**	
Description:	Identifier for *thatname*	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*thatid* > 0	
Name:	*thatname*	
Table:	**xtag**	
Description:	Key for *thatid*	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	

Name:	*thisdb*	
Table:	**xtag**	
Description:	Database account for the records specified by *thisname* and *thisid*.	
Format:	varchar2(32)	External: a32
NA Value:	- (hyphen)	
Range:	Any character string up to the column size that is a valid account name	
Name:	*thisid*	
Table:	**xtag**	
Description:	Identifier for *thisname*	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*thisid* > 0	
Name:	*thisname*	
Table:	**xtag**	
Description:	Key for *thisid*	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	Any character string up to the column size	
Name:	*time*	
Table:	**affiliation, alphasite**, **amplitude**, **arrival**, **bull_comp**, **calibrate**, **datacollected**, **datadays**, **detection**, **dlfile**, **interval**, **missed_class**, **origin (origin_ga), pixdisc**, **qcdata**, **qcstats**, **request**, **sensor**, **siteaux**, **timestamp**, **wfactivity**, **wfdisc (wfproto)**	
Description:	Epoch time, given as seconds since midnight, January 1, 1970, and stored in a double-precision floating number. Time refers to the table in which it is found [for example, in **arrival** it is the arrival time, in **origin (origin_ga)** it is the origin time, in **wfdisc (wfproto)** it is the start time of data, and in **siteaux** it is the start time for which measurements are valid]. Where the date of historical events is known, time is set to the start time of that date. Where the date of contemporary arrival measurements is known but no time is given, then time is set to the NA Value. The double-precision floating point number allows 15 decimal digits. At one millisecond accuracy, this is a range of 3 $*10^4$ years. Where the date is unknown or prior to February 10, 1653, *time* is set to the NA Value.	
Format:	float(53)	External: f17.5
NA Value:	NOT ALLOWED for **affiliation, arrival, calibrate, detection, interval, origin (origin_ga), sensor, siteaux, wfactivity, wfdisc (wfproto)** -9999999999.999	
Units:	Seconds	
Range:	*time* > -9999999999.999	

Name:	*timedef*	
Table:	**assoc (assoc_ga)**	
Description:	Time-defining code. This one-character flag indicates whether or not the time of a phase was used to constrain the event location. This column is defining (*timedef* = `d`) or nondefining (*timedef* = `n`).	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*timedef* ∈ {`n`, `d`}	
Name:	*timeres*	
Table:	**assoc (assoc_ga)**	
Description:	Time residual. This column is a travel-time residual measured in seconds. The residual is found by taking the observed arrival time (saved in the arrival table) of a seismic phase and subtracting the expected arrival time. The expected arrival time is calculated by a formula based on an earth velocity model (column *vmodel*), an event location and origin time (saved in **origin (origin_ga)** table, and the particular seismic phase [column *phase* in **assoc (assoc_ga)** table].	
Format:	float(24)	External: f8.3
NA Value:	–999.0	
Units:	Seconds	
Range:	*timeres* > –999.0	
Name:	*timesent*	
Table:	**msgdest**	
Description:	Time at which the corresponding message was sent	
Format:	float(53)	External: f17.5
NA Value:	-9999999999.999	
Units:	Seconds	
Range:	*timesent* > –9999999999.999	
Name:	*tlen*	
Table:	**ampdescript, dlfile**	
Description:	Time window length. If a velocity window is used, *tlen* should be NA in **ampdescript**.	
Format:	float(24)	External: f10.3
NA Value:	–1.0	
Units:	Seconds	
Range:	*tlen* > 0.0	
Name:	*toff*	
Table:	**ampdescript**	
Description:	Offset from theoretical or observed arrival time. This column is used to define the start time of the amplitude measurement window and may be used in conjunction with either *tlen* to define a static window or with *gvlo* to define a dynamic window. If *toff* is set to –999, then *gvhi* must be used to define the start time of the window.	
Format:	float(24)	External: f6.2
NA Value:	–999.0	
Units:	Seconds	
Range:	*toff* ≥ 0.0	

Name:	*topic*	
Table:	**datacollected, pixdisc**	
Description:	Type of data processing	
Format:	varchar2(8)	External: a8
NA Value:	NOT ALLOWED	
Range:	*topic* ∈ {AA , DA , IA , DF , BC , ED , EP}	
Name:	*transmeth*	
Table:	**msgdest**	
Description:	Method by which response is to be delivered to requestor.	
Format:	varchar2(16)	External: a16
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*tshift*	
Table:	**sensor**	
Description:	Correction for clock errors; designed to accommodate discrepancies between the actual time and numerical time written by data recording systems. Actual time is the sum of the reported time plus *tshift*.	
Format:	float(24)	External: f16.2
NA Value:	NOT ALLOWED	
Units:	Seconds	
Range:	Any floating point value	
Name:	*tsta*	
Table:	**ev_summary (an_summary, ex_summary)**	
Description:	Number of teleseismic observations for an event. A teleseismic observation is currently defined as having a station-event distance ≥ 2000 km.	
Format:	number(8)	External: i8
NA Value:	–1	
Range:	*tsta* ≥ 0	
Name:	*uncertainty*	
Table:	**netmag, stamag**	
Description:	Magnitude uncertainty. This value is the standard deviation of the accompanying magnitude measurement.	
Format:	float(24)	External: f7.2
NA Value:	–1.0	
Range:	*uncertainty* > 0.0	
Name:	*units*	
Table:	**amplitude**	
Description:	Units of amplitude measure	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	*units* ∈ {nm}	

Name:	*userid*	
Table:	**datauser, msgdisc**	
Description:	User identifier for Subscription and Message Subsystem	
Format:	number(8)	External: i8
NA Value:	–1 NOT ALLOWED for datauser	
Range:	*userid* > 0	
Name:	*username*	
Table:	**datauser, ftplogin**	
Description:	User name for Message Subsystem	
Format:	varchar2(24)	External: a24
NA Value:	– (hyphen) NOT ALLOWED for datauser	
Range:	Any character string up to the column size that is a valid user name	
Name:	*userstatus*	
Table:	**datauser**	
Description:	Status of user	
Format:	varchar2(24)	External: a24
NA Value:	NOT ALLOWED	
Range:	*userstatus* ∈ {ACTIVE, INACTIVE}	
Name:	*vamp*	
Table:	**amp3c**	
Description:	Vertical amplitude	
Format:	float(24)	External: f11.2
NA Value:	–999.0	
Range:	*vamp* ≥ 0.0	
Name:	*vang*	
Table:	**sitechan**	
Description:	Vertical orientation of seismometer. This column measures the angle between the sensitive axis of a seismometer and the outward-pointing vertical direction. For a vertically oriented seismometer, *vang* = 0 For a horizontally oriented seismometer, *vang* = 90 (see *hang*)	
Format:	float(24)	External: f6.1
NA Value:	NOT ALLOWED	
Units:	Degrees	
Range:	0.0 ≤ *vang* ≤ 90.0	

Name:	*verifstatus*	
Table:	**msgdisc**	
Description:	Status of verification	
Format:	varchar2(4)	External: a4
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*vmodel*	
Table:	**assoc (assoc_ga)**	
Description:	Velocity model. This character string identifies the velocity model of the Earth used to compute the travel times of seismic phases. A velocity model is required for event location (if phase is defining) or for computing travel-time residuals.	
Format:	varchar2(15)	External: a15
NA Value:	- (hyphen)	
Range:	Any character string up to the column size	
Name:	*vote*	
Table:	**discrimvote**	
Description:	The final vote for a given discriminant factor based on input from each station that contributed to that discriminant (see *discrim_flag*). For each origin that has gone through the classification process, there should be one vote for each discrimtype. The specific algorithms used to determine station votes for each discriminant are classified SECRET.	
Format:	varchar2(1)	External: a1
NA Value:	- (hyphen)	
Range:	*vote* $\in$ {1, 2, 3, 4, 5, 6, 7, A, B, C, D}	
Name:	*vsnr*	
Table:	**amp3c**	
Description:	Vertical signal-to-noise ratio. Ratio of *vamp* to root-mean-square amplitude of a vertically-oriented component filtered in a frequency band centered at *cfreq* Hz	
Format:	float(24)	External: f10.2
NA Value:	–999.0	
Range:	*vsnr* $\geq$ 0.0	
Name:	*wfid*	
Table:	**wfdisc (wfproto), wftag**	
Description:	Unique waveform identifier for a **wfdisc** record	
Format:	number(9)	External: i9
NA Value:	NOT ALLOWED	
Range:	*wfid* > 0	

Name:	*wgt*	
Table:	**assoc (assoc_ga)**	
Description:	Location weight. This column gives the final weight assigned to the allied arrival by the location program. This column is used primarily for location programs that adaptively weight data by their residuals.	
Format:	float(24)	External: f6.3
NA Value:	–1.0	
Range:	`0.0` < *wgt*	
Name:	*yield*	
Table:	**hydro_arrival**	
Description:	Sensor yield	
Format:	float(24)	External: f11
NA Value:	–1.0	
Units:	Kiloton	
Range:	*yield* ≥ `0.0`	
Name:	*ylderr*	
Table:	**hydro_arrival**	
Description:	Sensor yield error	
Format:	float(24)	External: f11.4
NA Value:	–1.0	
Range:	*ylderr* > 0.0	

Appendix C. View Descriptions

The **wfproto** view is the only view defined in the US NDC database schema. It contains the same columns as the wfdisc table, but is a view of the **calibrate**, **sensor**, **instrument**, and **wfconv** tables. It is used by the Data Acquisition Subsystem as a prototype for constructing the waveform descriptor records before they are written to the **wfdisc** table. Most of the columns in the view are inherited from the underlying tables, with the exception of *clip*, *commid*, *dir*, *dlfile*, *foff*, *nsamp*, *segtype*, and *wfid*, which are placeholders. The Data Acquisition Subsystem replaces these placeholders with actual values before the waveform descriptor record is inserted into the **wfdisc** table.

The table below lists the columns in the **wfproto** view and the tables from which they are inherited or the placeholder values for columns that are not inherited.

See Appendix B for column descriptions.

Table C1. wfproto

FIELD NUMBER	COLUMN	INHERITIED FROM	DESCRIPTION
1	*sta*	sensor.sta	Station code
2	*chan*	sensor.chan	Channel code
3	*time*	sensor.time	Epoch time of first sample in file
4	*wfid*	-1	Waveform identifier
5	*chanid*	sensor.chanid	Channel identifier
6	*jdate*	sensor.jdate	Julian date
7	*endtime*	sensor.endtime	Time + (nsamp-1)/samprate
8	*nsamp*	0	Number of samples
9	*samprate*	instrument.samprate	Sampling rate in samples per sec
10	*calib*	calibrate.calib	Nominal calibration
11	*calper*	calibrate.calper	Nominal calibration period
12	*instype*	instrument.instype	Instrument code
13	*segtype*	-	Indexing method
14	*datatype*	wfconv.outtype	Numeric storage
15	*clip*	-	Clipped flag
16	*dir*	-	Directory
17	*dfile*	-	Data file
18	*foff*	0	Byte offset of data segment within file
19	*commid*	-1	Comment identifier
20	*lddate*	sysdate	Load date

Appendix D. Accounts and Tables

The tables in Appendix D indicate which objects from Appendixes A and C are incorporated into the schema associated with each US NDC database account. A separate table is provided for each of the four US NDC databases.

Table D1. Summary of Unclassified Data Acquistion Database (OPSDB) Accounts and Tables

GLOBAL	LOOKUP
alphasite	affiliation
calibrate	chan_groups
channame	instrument
datauser	network
dlfile	sensor
dlman	site
ftpfailed	siteaux
ftplogin	sitechan
interval	stanet
lastid	wfconv
msgaux	
msgdatatype	
msgdest	
msgdisc	
request	
timestamp	
wfactivity	
wfaudit	
wfdisc (wfproto)	
xtag	

Table D2. Summary of Unclassified Archive Database (ARCHDB) Accounts and Tables

GLOBAL	LOOKUP
interval	affiliation
interval_files	chan_groups
lastid	instrument
wfdisc (wfproto)	network
	sensor
	site
	siteaux
	sitechan
	stanet
	wfconv

Table D3. Summary of Classified Processing Database (OPSDB) Accounts and Tables

AL1	AL2	DETPRO	DEVNULL	EVAL1
amp3c	amp3c	amp3c	amp3c	amp3c
amplitude	amplitude	amplitude	amplitude	amplitude
apma	apma	apma	apma	apma
arrival	arrival	arrival	arrival	arrival
assoc	assoc	detection	assoc	assoc
detection	detection		detection	detection
discrimuse	discrimuse		discrimuse	discrimuse
discrimvote	discrimvote		discrimvote	discrimvote
event_control	event_control		event_control	event_control
ga_tag	ga_tag		hydro_arr_group	hydro_arr_group
hydro_arr_group	hydro_arr_group		hydro_arrival	hydro_arrival
hydro_arrival	hydro_arrival		hydro_assoc	hydro_assoc
hydro_assoc	hydro_assoc		hydro_origin	hydro_origin
hydro_origin	hydro_origin		netmag	netmag
netmag	netmag		origerr	origerr
origerr	origerr		origin	origin
origin	origin		stamag	stamag
stamag	stamag			

Table D3. Summary of Classified Processing Database (OPSDB) Accounts and Tables (Continued)

EVAL2	FAL	GLOBAL	HAL	HYDRODET
amp3c	amp3c	alphasite	amplitude	amplitude
amplitude	amplitude	calibrate	apma	apma
apma	apma	channame	arrival	arrival
arrival	arrival	datauser	assoc	detection
assoc	assoc	dlfile	detection	event_control
detection	detection	dlman	event_control	hydro_arrival
discrimuse	event_control	event	hydro_arrival	hydro_origin
discrimvote	ga_tag	ftpfailed	hydro_origin	
event_control	hydro_arr_group	ftplogin	netmag	
hydro_arr_group	hydro_arrival	interval	origerr	
hydro_arrival	hydro_assoc	lastid	origin	
hydro_assoc	hydro_origin	msgaux	stamag	
hydro_origin	netmag	msgdatatype		
netmag	origerr	msgdest		
origerr	origin	msgdisc		
origin	stamag	remark		
stamag		request		
		timestamp		
		wfactivity		
		wfaudit		
		wfdisc (wfproto)		
		wftag		
		xtag		

Table D3. Summary of Classified Processing Database (OPSDB) Accounts and Tables (Continued)

LFDET	LOOKBACK	LOOKUP	MONITOR	MIGRATE
amp3c	amp3c	affiliation	an_summary	mig_date
amplitude	amplitude	ampdescript	bull_comp	mig_rules
apma	apma	aoi	datacollected	
arrival	arrival	chan_groups	datadays	
detection	assoc	colordisc	ev_summary	
	detection	gregion	ex_an	
	discrimuse	instrument	ex_summary	
	discrimvote	mapcolor	missed_class	
	event_control	mapdisc	pixdisc	
	ga_tag	mapover	qcdata	
	hydro_arr_group	mappoint	qcstats	
	hydro_arrival	network	station_hist	
	hydro_assoc	overlaydisc	station_type	
	hydro_origin	sensor		
	netmag	site		
	origerr	siteaux		
	origin	sitechan		
	stamag	sregion		
		stanet		
		wfconv		

Table D3. Summary of Classified Processing Database (OPSDB) Accounts and Tables (Continued)

RAL1	RAL2	REGDET	SOCCPRO
amp3c	amp3c	amp3c	assoc
amplitude	amplitude	amplitude	assoc_ga
apma	apma	apma	event_control
arrival	arrival	arrival	ga_tag
assoc	assoc	assoc	netmag
detection	detection	detection	origerr
discrimuse	discrimuse	origerr	origerr_ga
discrimvote	discrimvote	origin	origin
event_control	event_control		origin_ga
ga_tag	ga_tag		stamag
hydro_arrival	hydro_arrival		
hydro_origin	hydro_origin		
netmag	netmag		
origerr	origerr		
origin	origin		
stamag	stamag		

Table D4. Summary of Classified Archive Database (ARCHDB) Accounts and Tables

AL1	AL2	DETPRO	EVAL1	EVAL2
amp3c	amp3c	amp3c	amp3c	amp3c
amplitude	amplitude	amplitude	amplitude	amplitude
apma	apma	apma	apma	apma
arrival	arrival	arrival	arrival	arrival
assoc	assoc	detection	assoc	assoc
detection	detection		detection	detection
discrimuse	discrimuse		discrimuse	discrimuse
discrimvote	discrimvote		discrimvote	discrimvote
event_control	event_control		event_control	event_control
hydro_arr_group	hydro_arr_group		hydro_arr_group	hydro_arr_group
hydro_arrival	hydro_arrival		hydro_arrival	hydro_arrival
hydro_assoc	hydro_assoc		hydro_assoc	hydro_assoc
hydro_origin	hydro_origin		hydro_origin	hydro_origin
netmag	netmag		netmag	netmag
origerr	origerr		origerr	origerr
origin	origin		origin	origin
stamag	stamag		stamag	stamag
FAL	**GLOBAL**	**HAL**	**HYDRODET**	**LFDET**
amp3c	event	amplitude	amplitude	amp3c
amplitude	interval	apma	apma	amplitude
apma	interval_files	arrival	arrival	apma
arrival	lastid	assoc	detection	arrival
assoc	remark	detection	event_control	detection
detection	wfdisc	event_control	hydro_arrival	
event_control	wftag	hydro_arrival	hydro_origin	
hydro_arr_group	wftape	hydro_origin		
hydro_arrival		netmag		
hydro_assoc		origerr		
hydro_origin		origin		
netmag		stamag		
origerr				
origin				
stamag				

Table D5. Summary of Classified Archive Database (ARCHDB) Accounts and Tables (Continued)

LOOKBACK	LOOKUP	MONITOR	RAL1	RAL2
amp3c	affiliation	qcdata	amp3c	amp3c
amplitude	ampdescript		amplitude	amplitude
apma	aoi		apma	apma
arrival	chan_groups		arrival	arrival
assoc	gregion		assoc	assoc
detection	instrument		detection	detection
discrimuse	network		discrimuse	discrimuse
discrimvote	sensor		discrimvote	discrimvote
event_control	site		event_control	event_control
hydro_arr_group	siteaux		hydro_arrival	hydro_arrival
hydro_arrival	sitechan		hydro_origin	hydro_origin
hydro_assoc	sregion		netmag	netmag
hydro_origin	stanet		origerr	origerr
netmag	wfconv		origin	origin
origerr			stamag	stamag
origin				
stamag				

REGDET	SOCCPRO
amp3c	assoc
amplitude	event_control
apma	netmag
arrival	origerr
assoc	origin
detection	stamag
origerr	
origin	